National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

UNIVERSITY OF ALBERTA

REALISTIC HUMAN TRUNK SURFACES FROM UNORGANIZED THREE DIMENSIONAL DATA

BY

Roland B. Penner ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science.

DEPARTMENT OF ELECTRICAL ENGINEERING

Edmonton, Alberta
Spring 1996

National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-612-10745-0

Canada

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Roland B. Penner

TITLE OF THESIS: **REALISTIC HUMAN TRUNK SURFACES FROM UNORGANIZED THREE DIMENSIONAL DATA**

DEGREE: Master of Science

YEAR THIS DEGREE GRANTED: 1996

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

(Signed) . . Roland Penner . . . . . .

Roland B. Penner
Box 427
Tofield, Alberta
Canada, T0B 4J0

Date: Jan. 23, 1996. .

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **REALISTIC HUMAN TRUNK SURFACES FROM UNORGANIZED THREE DIMENSIONAL DATA** submitted by Roland B. Penner in partial fulfillment of the requirements for the degree of Master of Science.

Dr. Nelson Durdle (Supervisor)

Prof. Art Peterson

Mr. Jim Raso

Dr. Keith Stromsmoe

Date: Jan. 23, 1996.

Dedicated to those who rule.

ABSTRACT

The problems of displaying an irregularly shaped surface created from noisy data manifest themselves in unrealistic and inaccurate surface characteristics. These problems are relevant to medical visualization in which the surface data must appear as close to the actual surface as possible. It is discussed in detail how to transform three dimensional discrete points from the surface of a human trunk into a realistic and accurate surface visualization. Realistic human back surfaces were created that can be used in the assessment of subjects with scoliosis. Methods are presented which minimize the effects of noise and create realistic solid surfaces that can be viewed and manipulated in six degrees of freedom. Automated techniques of boundary detection, triangulation and noise reduction are discussed. This work determines the best techniques for transforming three dimensional back surface data, with the criteria being: realism of presentation and surface accuracy. Specifically, a linear triangulation algorithm is developed which defines the data boundaries. A subsequent non-linear filtering of the triangulated data produces the desired noise reduction effects.

ACKNOWLEDGEMENTS

I acknowledge the many scientists and engineers who take pride in their work and provide the coming generation solid evidence of their progress and the means to take the next step.

I am thankful for the funds made available for research by the Glenrose Rehabilitation Hospital and the National Research Council.

Thanks:

For the brainstorming and reflective powers Ketan Bhalla, Daryl Maier and Michael Polak as I would bounce ideas off them. Some of the returns were over my head, some were off the wall, but there were a few that scored and I am grateful.

For the sounds of music which resonate in my mind. The peaceful and uplifting sounds from Hausmusik, Moser's groups and Taize.

No thanks to ID software, creators of DOOM.

A big number 10 Eskimo burp to all relatives: A.K.A. eat and run restaurants and moral supporters.

I finally thank my entertainment and sports coordinators Michael Polak, Daryl Maier, and Nick Bucyk.

TABLE OF CONTENTS

1 Introduction **1**

1.1 Objectives / Scope 1

1.1.1 Motivation 1

1.1.2 Computer Environment 3

1.1.3 Representation Criteria 4

1.2 Background 4

1.2.1 Scoliosis 4

1.2.2 Data Acquisition 7

1.2.3 Problems to Overcome 12

1.3 Thesis Organization 13

2 Literature Review **14**

2.1 Boundary Generation 14

2.1.1 Qualification / Assumptions 15

2.1.2 Boundaries through sectioning 15

2.1.3 Boundary Generation After Triangulation (Graphing Theory) 17

2.1.4 2-D Veltkamp 18

2.1.5 3D Veltkamp 19
2.2 Triangulation 24
2.2.1 Trimming the Delaunay Triangulation 29
2.2.2 Point Density based Triangulation 29
2.3 Noise Reduction 32
2.3.1 Noise Reduction through Equation Fitting 34
2.3.2 Resampling to Organized Points 35
2.3.3 Filtering 35
2.3.4 Summary 36

3 Boundary Formation **37**
3.1 Background 37
3.2 Implementation 38
3.2.1 Density 39
3.3 Maximum Line Length 41
3.4 Further Considerations 45
3.4.1 Summary 46

4 Triangulation **48**
4.1 Initial Triangulation (Stage 1) 50
4.1.1 Triangulation about one Point 52
4.2 Triangle Optimization (Stage 2) 57
4.2.1 Summary 60

5 Noise Reduction **61**

5.1 Resampling 61

5.2 Standard Filters 63

5.2.1 Implementation 64

5.3 Custom Filter 65

5.3.1 Implementation 66

6 Results **67**

6.1 Accuracy 67

6.1.1 Accuracy - Boundary 67

6.1.2 Accuracy - Triangulation 68

6.1.3 Accuracy - Effects of Noise 73

6.2 Realism 73

6.2.1 Realism - Boundary 73

6.2.2 Realism - Triangulation 75

6.2.3 Realism - Effects of Noise 78

6.3 Speed 81

6.3.1 Speed - Triangulation 81

6.4 Automation 82

7 Conclusions and Future Work **85**

7.0.1 Future Work 87

Bibliography **91**

LIST OF FIGURES

1.1 Spinal Curves 6

1.2 Single projector/Single camera setup 7

1.3 Structured Light Patterns: A dorsal patient view 8

1.4 Line Rasterstereography 9

1.5 Data distribution from a uniform grid of dots projected onto a back surface 10

2.1 Ambiguous boundaries 16

2.2 Two dimensional Veltkamp 19

2.3 Shaded test data for Veltkamp's boundary generation 20

2.4 Boundary Erosion 22

2.5 Three dimensional Veltkamp 23

2.6 Triangle Shape Comparison: Sliver vs Equilateral 27

2.7 Delaunay circle 28

3.1 Point Classifications 39

3.2 Density Approximation Graph 41

3.3 Density Histogram 42

3.4 Connection Lengths Graph 43

3.5 Rectification and Rotation 47

4.1 Triangulation Partitioning 51

4.2 Sample Point Neighbourhood 52

4.3 Triangle Interior Angles 54

6.1 Boundary Accuracy: The computer generated boundary is shown in the black outline 69

6.2 Triangle Mesh and Holes 71

6.3 Inappropriate Object for Triangulation 76

6.4 Triangle Optimization 77

6.5 Triangle Shading 78

6.6 Back Surface I: Noise Comparison 79

6.7 Back Surface II: Noise Comparison 80

6.8 Total Triangulation Time vs Number of Points 83

6.9 Triangulation and Initialization Graph 83

6.10 Back Surface I: different view points 84

7.1 Graphical Summary 88

LIST OF TABLES

4.1 Creating one Triangle 52

6.1 Surface Accuracy 72

6.2 Triangulation Algorithm Processing Time (Orders) 82

Chapter 1

Introduction

1.1 Objectives / Scope

The objective of this study was to develop a data transformation which will convert discrete three dimensional data into a realistic and accurate representation of the human back. In doing so, the process should minimize user interaction and errors due to data transformation, and maximize the speed of the computer representation and manipulation.

1.1.1 Motivation

To better understand the effect of spinal misalignment on distortion of the human trunk, tools are needed to measure and quantify surface features. In cooperation with the Glenrose Rehabilitation Hospital, back surface modeling was performed to aid in the study and treatment of scoliosis. A program was developed which allows clinicians to compare and analyze digital images, three dimensional spinal column data, and

three dimensional back surfaces. This work contributes to the back surface portion by combining image processing and data visualization techniques to create accurate and realistic three dimensional computer generated models of the human back from discrete, three dimensional points.

Clinicians require a means of visualizing the trunk deformity in three dimensions, because a three dimensional model adds to the number of physical measures that can be made on a back surface from a patient in the standing position. A set of data transformations is required to manipulate discrete data points into a computer representation which approximates the actual back surface. The model should be realistic in order to aid a clinician in deciding on a treatment protocol or assess the effects of treatment. The data contains only shape information, because color and texture are not acquired. Features of interest include: the asymmetry of the entire trunk; the general shape and directions of the spinal curvature; the asymmetry of the shoulders; and the narrowing or widening of the trunk at the waist or various heights.

Many of the features of interest are not easily definable in term of mathematical equations or absolute positioning, so an appropriate definition of a feature is now introduced.

> *Minimum* **feature** *size is defined by sampling frequency. The highest sampling frequency defines the smallest feature that can be detected by the data acquisition system. Since the currently used system has a distance between samples of 1 cm, the features that can be detected are, by Nyquist's sampling theorem [5], just over 2 cm in diameter.*

Any representation of the data should make it possible to single out features of the minimum size. How this information is portrayed will determine the quality of the representation. The quality (realism) will be judged against pictures of the patient and accuracy is determined by how much the resulting surface deviates from the original data points.

1.1.2 Computer Environment

The protocol for data transformations was developed for use on an IBM RS6000 computer with 32 Megabytes of random access memory (RAM). The computer was equipped with a graphics accelerator card with a depth buffer and a three dimensional matrix pipeline for image rotation, translation and scaling. Also built into the graphics card are the equations for simple lighting and shading calculations. Any method developed was required to work at reasonable speeds on this computer; on the order of minutes (0-5 minutes). This is fast enough to preprocess the maximum number of data sets acquired per day. The on-screen rendering of the surface data had a time constraint on the order of seconds (0-5). The ability to view the back surface from many different view points requires this short time period for rendering to make it a viable tool for use in a clinic.

The low level software routines were all written using C or using the IBM Graphics Language (GL). The interfacing was written using the X and Motif libraries and a good base for the interfacing was written by various other programmers prior to this study.

There are a variety of possible approaches to creating a realistic surface, all of which depend on the size of the data set and the resources available. The best methods were sought for only the given computer resources and the existing back surface data sets.

1.1.3 Representation Criteria

In general, a realistic and accurate representation of the back surface is desired. This representation must be easily manipulated by a clinician. The ideal representation would have the following characteristics:

1. all the visible properties of skin;
2. the shape of the actual and computer representations look identical to the human eye;
3. a surface which lies strictly within the boundaries of the data;
4. a fully connected solid surface; and,
5. instantaneous rotation and on screen manipulation of the surface.

1.2 Background

1.2.1 Scoliosis

Scoliosis is an abnormal lateral curvature of the spinal column coupled with rotation of the individual vertebra. Some cases of scoliosis (about 20%) are secondary to other

conditions such as spina bifida or muscular distrophy, or are the direct result of known events such as radiation or malnutrition. The majority of scoliosis (about 80%) falls under the category of idiopathic scoliosis. This means that the spinal curvature occurs for no known reasons.

Scoliosis is usually detected during the adolescent growth spurt. As the spine grows, the spinal column begins to bend and twist. A non-scoliotic spine appears straight when seen from the dorsal view (figure 1.1 A). As seen from the side, a normal spine has a rounding in the upper part of the back (thoracic kyphosis) and a depression in the lower part (lumbar lordosis). With scoliosis, (figure 1.1 B) the spine is 'S' shaped in the dorsal view. Most scoliotic curves bend to the right in the upper part of the back and have a compensating curve to the left in the lower part of the back. The rib cage, pelvis and spinal column can all be affected. The asymmetries can cause back pain, physical problems such as imbalance, an awkward gait and respiratory problems. The awkward appearance of a scoliotic patient causes negative reactions from the self-conscious patient as well as other people. Many people with scoliosis seek treatment because they are unhappy with their appearance.

The current alternate treatment methods are bracing for moderate scoliosis and surgery for severe cases. Braces are placed around the whole trunk with the intent to apply corrective forces to the trunk. Surgery is used to support the structurally unstable spine and to correct as much as possible the asymmetries of the trunk. Pre-bent rods are fixed to the spine with wires, hooks or screws to support and mold the spine. The effectiveness of these treatments is hard to assess. A consistent

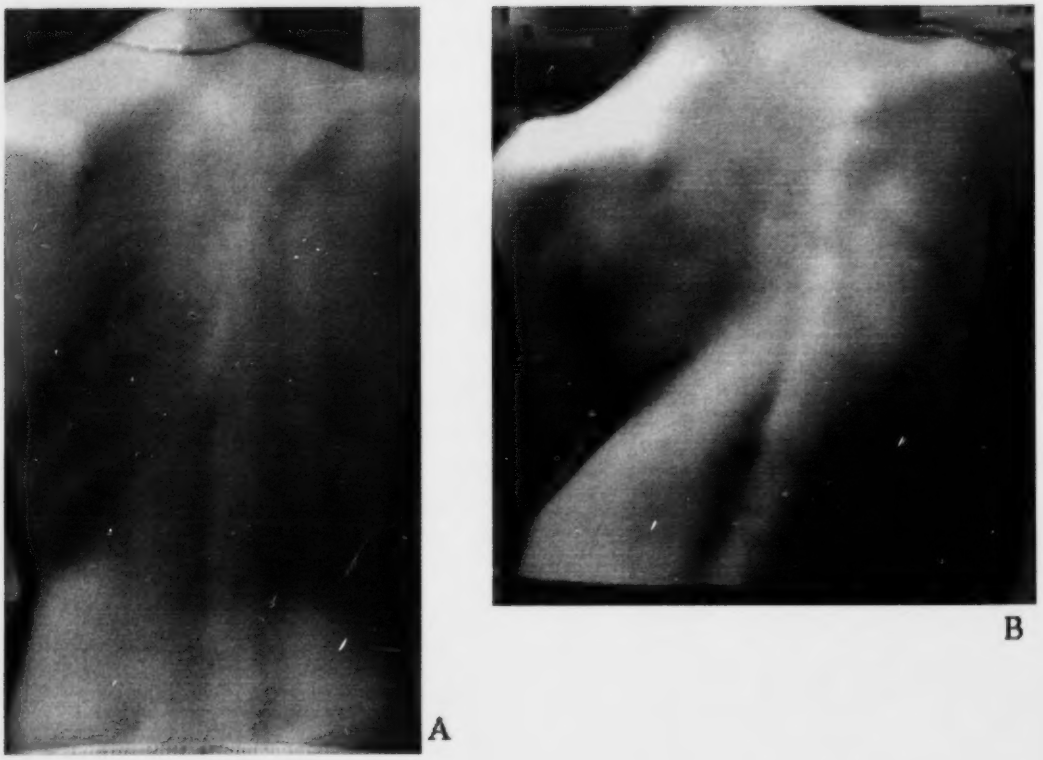

Figure 1.1: Spinal Curves: A) Normal dorsal (back) view; B) Scoliotic dorsal (back) view

quantification of the changes due to each treatment would help in this assessment and is one of the goals of future study.

1.2.2 Data Acquisition

Back surface data is obtained from the scoliosis clinics at the Glenrose Rehabilitation hospital. The acquisition uses one camera and a projected light pattern. A standard 35mm projector is used. The setup is shown in figure 1.2. The back is essentially featureless, so features are imposed by projecting a light pattern upon the back. The depth of the surface is calculated by matching the imaged light pattern to that of a reference image. The reference image is a camera image capture of the light pattern projected onto a vertical plane.

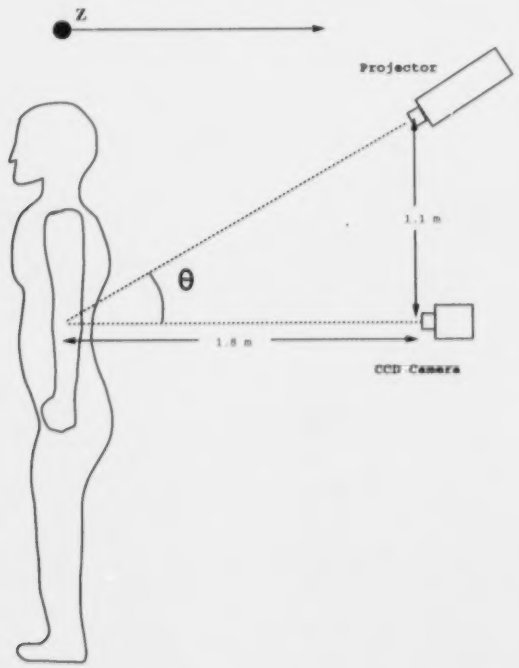

Figure 1.2: Single projector/Single camera setup

The projected light pattern is a regular grid of points. When projected it results in white points of light against a dark back surface (figure 1.3). One image is taken of the points on the back surface and the second is taken of the points projected onto a vertical flat board (reference grid) located at Z=0. In practice, the reference grid remains constant and the reference needs to be re-imaged only when the system layout is changed or inadvertently jarred. The light patterns between the two pictures need to be matched in order for the third dimension to be calculated.

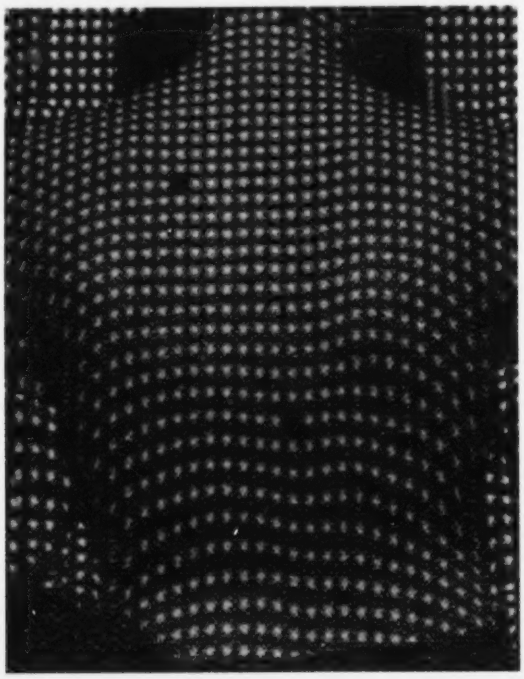

Figure 1.3: Structured Light Patterns: A dorsal patient view

An automated data acquisition system [18] has been developed which will replace the currently used method. The method is called line rasterstereography. The dot pattern is replaced by a horizontal line pattern and the spacing between lines is comparable to that of the vertical spacing of dots. Two cameras are situated with

one below and one above the projector. There is no difference in the format of this new data (figure 1.4), therefore the methods of data transformation will apply to these data sets as well.

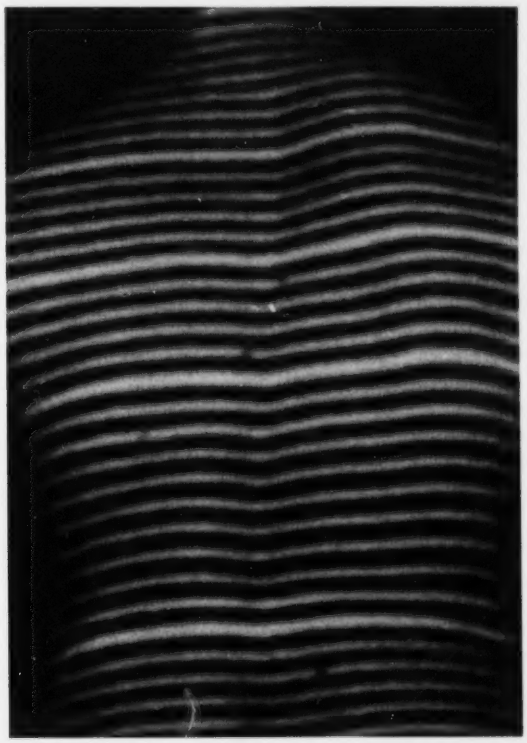

Figure 1.4: Line Rasterstereography

The data is a set of XYZ triplets. The most significant error in the data is in the Z component and it is less than ±2mm [6]. The points are randomly ordered with a semi-uniform density. A sample of the data distribution from a dot projection is shown in figure 1.5. Although the projector generates a uniform pattern the camera sees the points of light at an angle, thus the distorted pattern. The planar density of the points is approximately one point per square centimetre. For other light patterns the density changes depending on the period of the light pattern and the camera

angles. Since the exact density of the data and the pattern of the data is not known, it is assumed that the points are randomly distributed over the surface in a relatively uniform density.

Figure 1.5: Data distribution from a uniform grid of dots projected onto a back surface

Data capture does not distinguish folds (occluded regions) or sharp creases (cusps) in the surface. Therefore, the points are assumed to represent a continuous surface. Coverage of the entire back surface from the neck to the hips is acquired. The number of points in a data set varies between 500 and 2000. This number depends on the density of points and the size of the child that is being imaged. Only the Cartesian coordinates of these points are stored, but not in any particular order. So from the computers point of view there is a random distribution of data points

The data will be connected into a solid triangular mesh for the final representation. The triangles would be Gouraud shaded [4] in an artificially lit environment. Justification for this choice comes from the criterion for the final surface. This representation would be a solid surface and could be fully connected. The simplest surfacing element is a polygon and the simplest polygon is a triangle. As a result of these two facts, the hardware can reproduce the surface at the fastest possible refresh rate for a solid object. The combination of lighting and shading produces a reflectance and shading that closely resembles that of the actual surface in a lit room. The lighting model would need to be flexible to accurately depict different skin pigmentations.

Other possible methods of surface representation are not as well suited for the application or have additional constraints put on the data. Surface splines are significantly affected by noise in the data, and the ordering of data points must be known in order to generate the surface. This and the lengthy computations required to render the spline surface made this option unrealistic for clinical viewing. It takes on the order of 5-10 seconds to render wire splines and 10-20 seconds to render spline surfaces. Wire patch models render quickly, but they still require organized points to calculate higher order curves. The wire models do not interpolate over the whole surface and therefore do not provide the illusion of a solid surface. If rendering time is to be optimized and a solid surface created, a triangular mesh is required to display the surface.

1.2.3 Problems to Overcome

The second criterion, (i.e. the shape of the actual and computer representations look identical to the human eye), is limited by the density of the data. The decision to use triangles keeps the surface shape and structure as localized as possible. This means that the fewest number of points in the smallest area can be connected to form a surfacing element. It is non-trivial to define the triangle connections and the problem is compounded when combined with the fifth criterion where connections must lie within the boundary of the data.

Aspects of the data that are directly opposed to the development of an accurate and realistic surface are the random distribution of points, the lack of a defined boundary and the noise present in the data. The random distribution of points makes it harder to define a neighbourhood of points, because there is no means of predicting where any point will lie. Point neighbourhoods are necessary for connecting the data points into the desired triangles.

The lack of a boundary leads to ambiguity when defining where the surface begins and ends in three dimensional space. This could lead to defining triangles that overlap or exist outside the surface. These connections would in turn appear as false surfaces and confuse analysis. Finally, noise in the data may either mask features or create undesirable false features.

The categories of data transformations which constitute the possible means to accomplish the objectives, and that will be further investigated are listed below:

1. a process of connecting data points into a triangle mesh (triangulation). Trian-

gulation is required to create a fully connected surface of triangles;

2. a boundary generation process, to constrain the connection of points to within the region of interest;

3. a process to reduce the effects of noise, so the representation does not exhibit false features;

4. a means to display the data after all preformatting and connection definitions are complete; and,

5. standard noise reduction techniques require the data to be ordered, therefore a resampling or reformatting technique is required.

1.3 Thesis Organization

The remainder of the thesis will concentrate on the three main topics: boundary generation, triangulation and noise reduction. Each topic is reviewed separately in chapter 2. The methods of implementation are then presented in detail in chapters 3 through 5. These methods are then evaluated in chapter 6, using the criterion set out in the objectives. The results presented in chapter 6 are accompanied by pictures of various surface representations at different stages of the data transformation. Each portion of the data transformation will eventually mesh with the others to provide a comprehensive means for a realistic data visualization.

Chapter 2

Literature Review

2.1 Boundary Generation

There are few applications that require boundary generation techniques. Most data sets have boundary information stored within the data. Applications like Magnetic Resonance Imaging and volume visualization [7][23][9][2] will most often have the data for visualization come in the form of a regular grid of data points. Other applications will have regular sampling or simple boundaries [8][16][11], which implicitly define the boundary. All boundary points in these applications lie in set positions or are already defined to be part of the boundary polygon. This has limited the amount of research that has been done on the subject of boundary generation for irregularly shaped objects. For the case of the human trunk, the boundary is ambiguous; knowledge of the actual boundary shape is necessary to generate an acceptable boundary.

2.1.1 Qualification / Assumptions

It is clear that a boundary can be defined manually by someone with knowledge of the original object described by the data. This discussion will focus on techniques of boundary generation that minimize user interaction.

The boundary was sought in two dimensions (in the XY plane, disregarding Z (depth)). A boundary will be defined as a cyclic graph of the outermost data points. The interior points need to be excluded from the boundary and the boundary should encase all interior points with a single closed polygon. This excludes the need for recognition of holes within the interior of the surface.

For an irregularly shaped boundary, as in this case for the back data, it is desirable to have a uniform density of data points that uniquely and adequately define a boundary. This is because multiple boundaries can be derived from identical data sets (figure 2.1). A smooth (coarse) boundary or a jagged (fine) boundary can both be generated from the same data sets. The best boundary was one that was indicative of the underlying surface. Any method of boundary generation relies on knowledge relating to data density and the underlying object itself in order to find an appropriate bounding polygon.

2.1.2 Boundaries through sectioning

A method of sectioning the data points into a grid of regions was developed by Zhang [28]. If points are found within a grid region, the extreme points in the vertical and horizontal (away from the center of the data set) are taken as being boundary points.

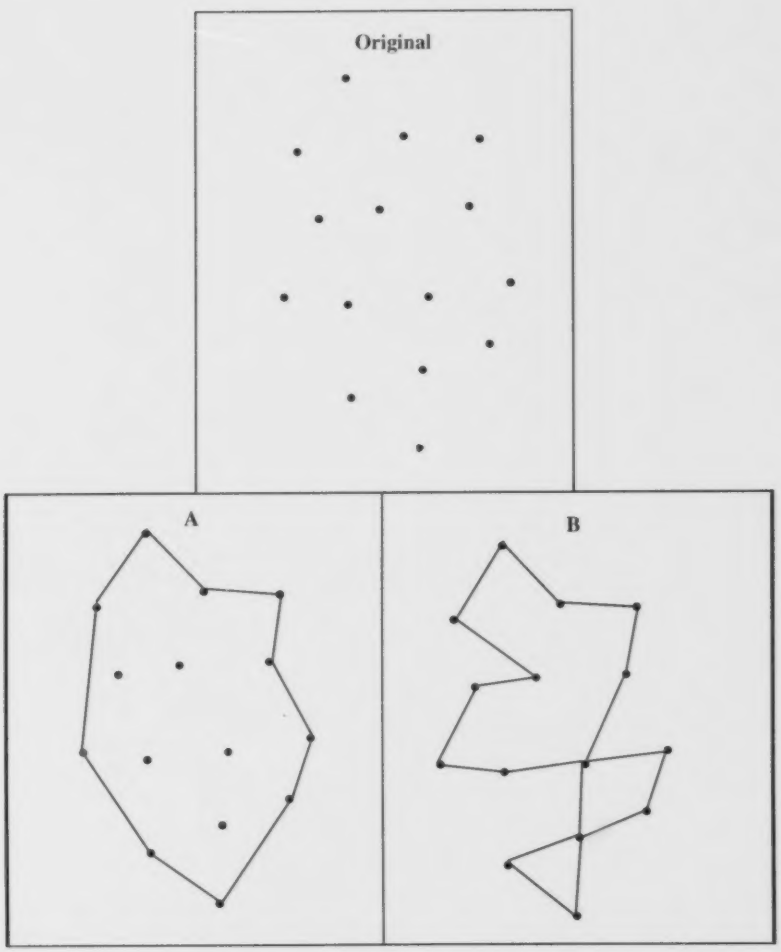

Figure 2.1: Different boundaries for the same data set: A) Course boundary; B) Fine boundary

Either, one point has both the extreme value for the X and Y coordinate, or multiple points, each an extreme in one direction are chosen as boundary points.

The order of surface traversal is important as well as choosing the optimal size of grid. Choosing a grid that is too small can lead to boundaries crossing one another [28] and a grid that is too large causes regions outside the data set to be included within the boundary. The way in which the final boundary is realized is highly dependent on the grid size chosen, or correspondingly, is highly dependent on the density of the points in the data set.

User interaction is necessary, but undesirable, to define the grid size that is appropriate for the data set. Even with an appropriate grid size, due to the irregular point distribution, the boundary may still cut through corners leaving valid data points outside the boundary. By the same reasoning, the boundary may enclose portions which are not part of the region of interest. An optimal choice of grid spacing can minimize these negative effects, but is not guaranteed to eliminate them [28].

2.1.3 Boundary Generation After Triangulation (Graphing Theory)

A less obvious approach is taken by Veltkamp [26]. When triangulating an irregularly shaped data set it is necessary to define a boundary to ensure that triangle connections do not degrade the integrity/continuity of the surface. In Veltkamp's thesis (published in book form as given in reference [26]) triangulation is performed ignoring all criterion for a boundary. This approach generates an excess of false triangles. The way in

which a boundary is created is to systematically remove those triangles which defy the criterion set out for the boundary of the object. The data sets that are used in his discussions produce boundaries that are Hamilton cycles [26]. This means they are both cyclic and contain all the data points. Cyclic means that there is at least one path that goes in a circle through at least 3 unique points. However, the interior points are strictly not to be included in the two dimensional boundary for the data sets of interest here. Instead, all valid interior points should be contained within the bounding polygon.

2.1.4 2-D Veltkamp

Veltkamp [26] generalizes his boundary generation to N dimensions and the criterion for triangle elimination in two and three dimensions are of interest.

The process that is followed for boundary generation is illustrated in figure 2.2. An initial outer boundary is present after the triangulation. The triangle edges along this boundary each see an angle opposite themselves. The triangles with the largest angles become candidates for removal. The interior angles which are the largest (obtuse) are removed first. Removals are made until the boundary contains all data points. As will be discussed later this will not necessarily work for data distributions containing non-boundary points. Triangles are restricted from being removed if all three points of the triangle already lie on the boundary. This ensures that all triangles within the interior of the boundary are not removed, since these triangles all have three boundary points as vertices. It also ensures that the boundary is a single cyclic graph. Once

all triangles have all three points on the boundary the triangle removal terminates. Veltkamp produced boundaries which closely represent the true boundaries of his test objects figures 2.2 and 2.3.

Figure 2.2: Two dimensional Veltkamp [26]: 4 stages of boundary generation
A) Original points; B) Delaunay Triangulation; C) The desired boundary (bold lines), extraneous connections (dashed lines); D) Outer triangles have been Eliminated

2.1.5 3D Veltkamp

The simple difference when generalizing Veltkamp's methods to three dimensions is that a tetrahedron's interior angles opposite the initial three dimensional boundary is used for the removal of triangles.

The three dimensional examples given, assume that the surface is a closed three dimensional object. This produces triangles across the back of the surfaces. The

Figure 2.3: Shaded test data for Veltkamp's [26] boundary generation: A,C) Final boundary (triangle mesh); B,D) Bezier triangle subdivision and shaded surface (continuous over the first derivative)

undesirable effect is these triangles may pierce portions of the mid-surface as they stretch from edge to edge across the back. Triangles stretching from side to side across the back of the facial mask can be seen in 2.3 C).

Veltkamp recognized the need for boundary generation, because of the many depth maps of data produced in computer vision and geometric modeling. However, all these methods initially capture data in the form of a single depth map. Veltkamp's methods do not take advantage of the added information gained by knowledge of the data acquisition format. The fact that there will be no overlapping of points, and that there is a uniform distribution of data would be helpful in ensuring that triangles do not stretch across the backside of the surfaces, or potentially through the surface

The two dimensional version of his boundary detection is one which will eliminate the extraneous triangle surfacing across the back of the data set.

However, it is unclear without further investigation whether it would be possible to use this method on a two dimensional data set in which there are points within the boundaries that are not meant to be incorporated into the boundary. It is possible that the process of eroding triangles may not have a sufficient terminating condition. If there are internal points to a two dimensional boundary, it is foreseeable that the boundary could erode the actual boundary and cut inwards and appear as snake like paths within the surface (figure 2.4).

Veltkamp's work was useful in that it illuminated the problems of triangulation across the back of three dimensional surfaces. It also provided further support for using the Delaunay triangulation process. Veltkamp is the only person who has used a

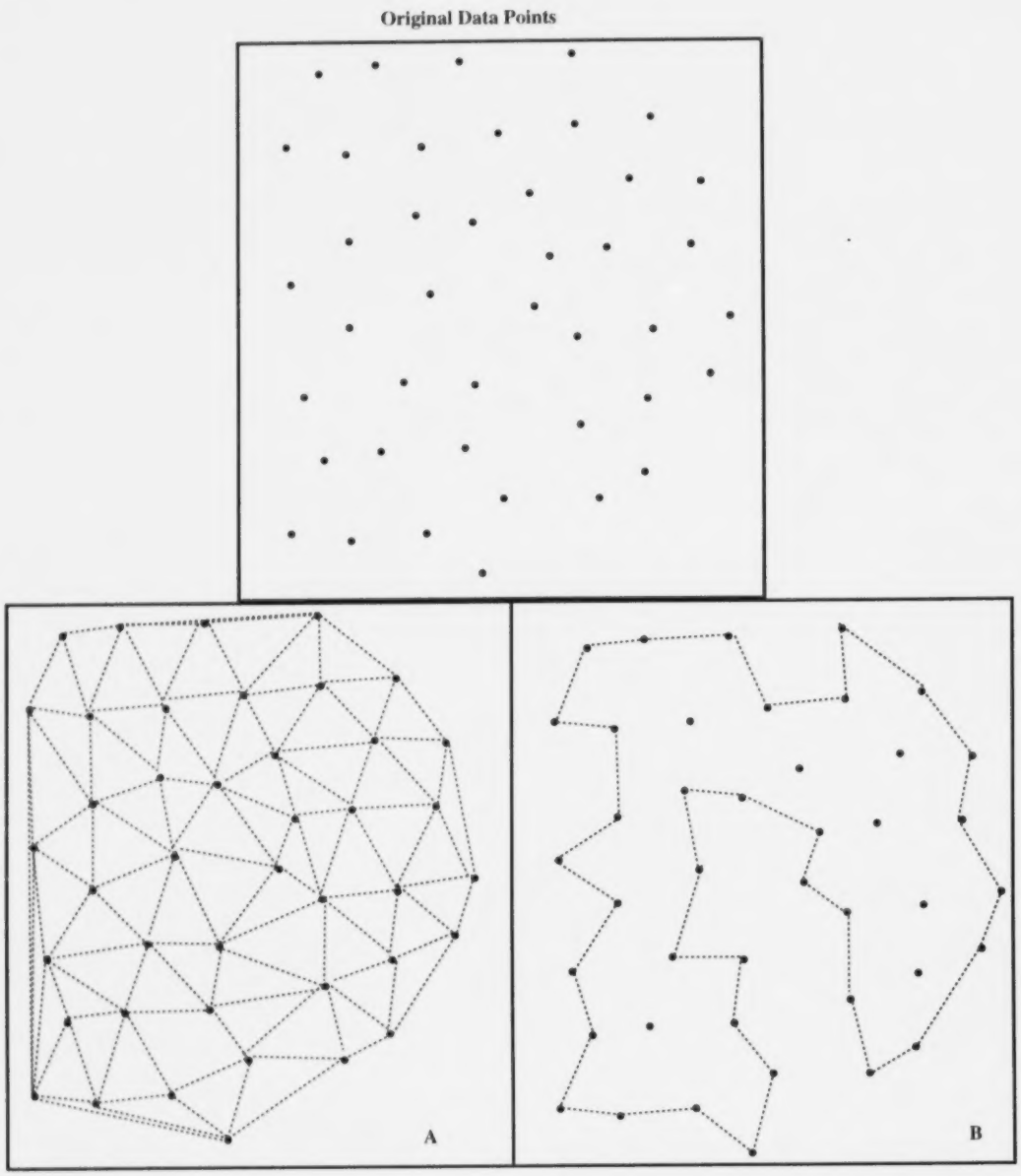

Figure 2.4: Boundary erosion: A) Delaunay Triangulation; B) Possible erosion of boundaries

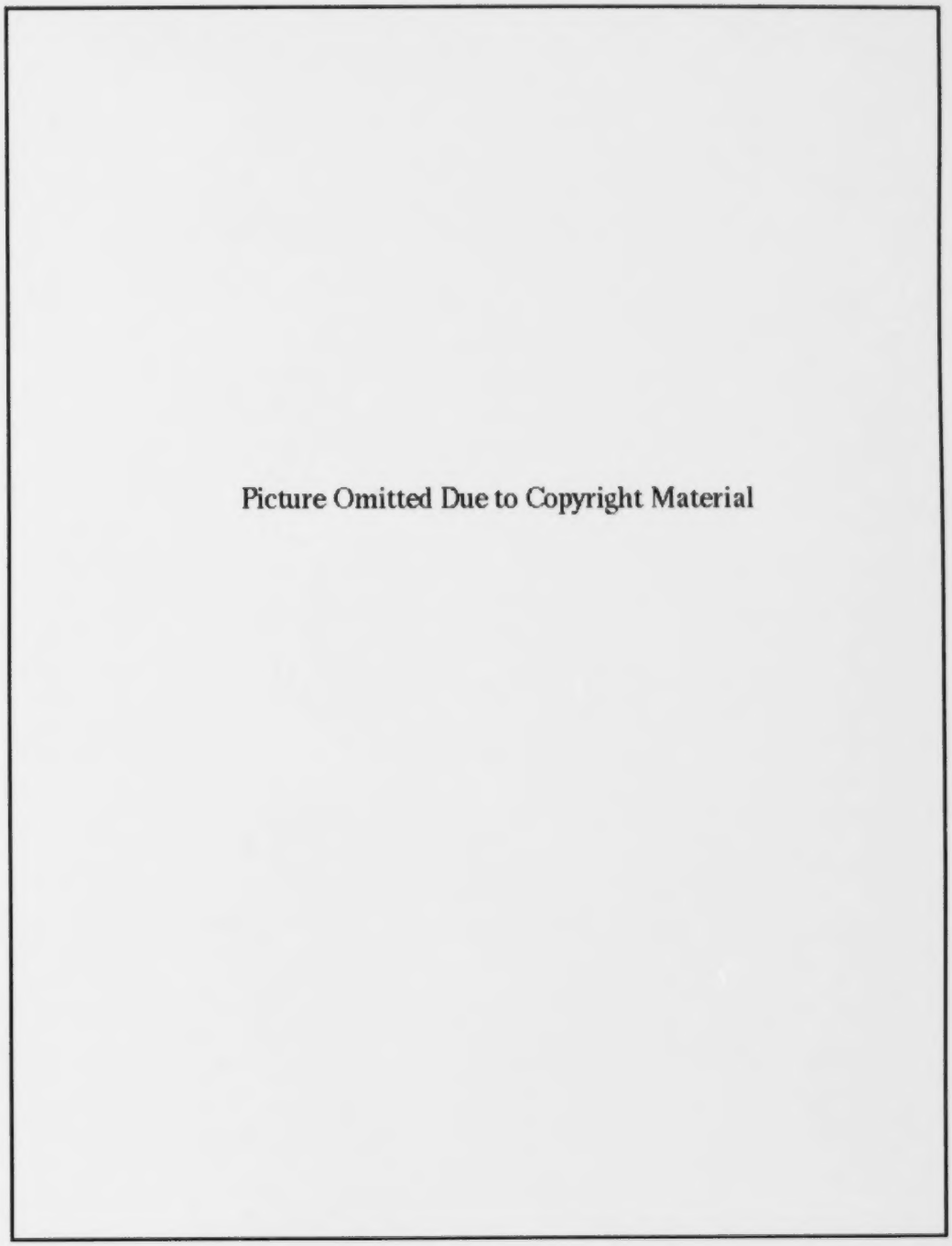

Figure 2.5: Three dimensional boundaries by Veltkamp [26]: The successive stages of boundary generation for a mask reconstructed from laser-range data points (1468 vertices and 2930 triangles). Bottom right: wire frame of the final boundary.

triangulation technique in order to define an outer boundary. Delaunay triangulation is used to create a surface and then the obtuse triangles are removed from the boundaries. This has a similar effect as minimizing the number of obtuse triangles from a triangulated surface, which is the basis of the implementation to be presented here. The stages of the triangulation are depicted in figure 2.5 and are further explained in the next chapter.

2.2 Triangulation

The simplest surfacing polygon is a triangle. The triangle produces a planar (bilinear) interpolation between the discrete points. The Delaunay triangle (described later in this section) is also the most local connection that can be made between data points. Other surfacing polygons like square patches and circles are less efficient at surfacing when compared to the triangle, because the graphics hardware was optimized for triangle generation and triangle shading. A Non-Rational B-Spline (NURBS) surface was also considered for surfacing, but it renders an order of magnitude slower than single triangles. While a single triangle mesh takes one second to render the corresponding NURBS surface takes ten seconds. NURBS is implemented with many small triangles fit to a spline in two dimensions. These smaller triangles are a close approximation to the spline curve and create a continuous surface over the first derivative. As will be further described in the discussions on noise, splines are not well suited to data with noise. The problems with noise in the data and the slow rate of rendering made NURBS an unrealistic choice for the surface representation.

Triangulation is a well studied field for irregular point distributions. A triangle has been a very convenient means of representing surfaces and volumes of data on computers [7][9]. The coordinates of only three points in three dimensional space define a plane and its normal. The plane bilinearly interpolates between the three points to define a solid surface element.

Most techniques for triangulation come to a point where they have a quadrilateral defined [13]. The four points are in the smallest possible neighbourhood. A decision is required as to which of the two diagonals are to be used to make the two triangles. There are at least three ways to decide upon the diagonals. One is to minimize the diagonal length so that all connections are as short as possible [13]. The next is to maximize the minimum interior angles in the triangles [13][22] so that surface area is maximized. The last is to minimize the maximum interior angle which attempts to eliminate obtuse angles [13].

The minimizing the diagonals and minimizing the maximum interior triangle angles may create slivers (thin triangles) which produce a non-continuous illusion in a triangulated surface (figure 2.6). The same effect can occur by minimizing the maximum interior angles. Slivers are caused by choosing a triangle with a minimized area or small interior angles. The reason for the non-continuous illusion is due to the calculation of the surface normals. Surface normals which are very different on adjacent surfaces creates a sharp contrast in lighting. If the long, thin triangle connects points which are far apart (i.e. not in the most local neighbourhood), then the Z coordinates can vary with a greater magnitude than with local triangle connections. The large

changes in Z across a single triangle result in sharp contrast in light shading across the triangle.

Lawson's [22] maximizing the minimum angle is equivalent to the Delaunay triangulation [1]. By maximizing the triangle's interior angles the triangles are kept as close to equilateral as possible; subsequently the average area of the set of triangles is maximized [22][11]. This is the only method that has a unique solution that can be found without testing every set of triangles [13]. The speeds of a maximizing the minimum angle and the Delaunay algorithm is greater than any other method of triangulation for irregular data sets. The reason for this is the Delaunay algorithm is the only non-exhaustive triangulation. The two dimensional Delaunay is O(nlogn). There is also a three dimensional equivalent, but it can be shown that a two dimensional triangulation is equivalent [3], given the data does not fold back on itself. The back surface data is a depth map and cannot describe folds in the surface, so the triangulation can be performed in the XY plane, without loss of information.

The Delaunay triangulation superimposes a circle that circumscribes the three points of a potential triangle (figure 2.7). If no other point in the data set lies within that circle it is considered the best choice for a triangle [22]. In the case where a point lies within the circle (point 'D' in figure 2.7) a more optimal triangle can be created; specifically, a triangle with the vertices 'B', 'D' and 'C'. Subsequently, when a circle circumscribes this new triangle it will not have point 'A' in its interior.

NOTE: From here on the only criterion that will be used for choosing an optimal triangulation will be the Delaunay circle criterion, which is

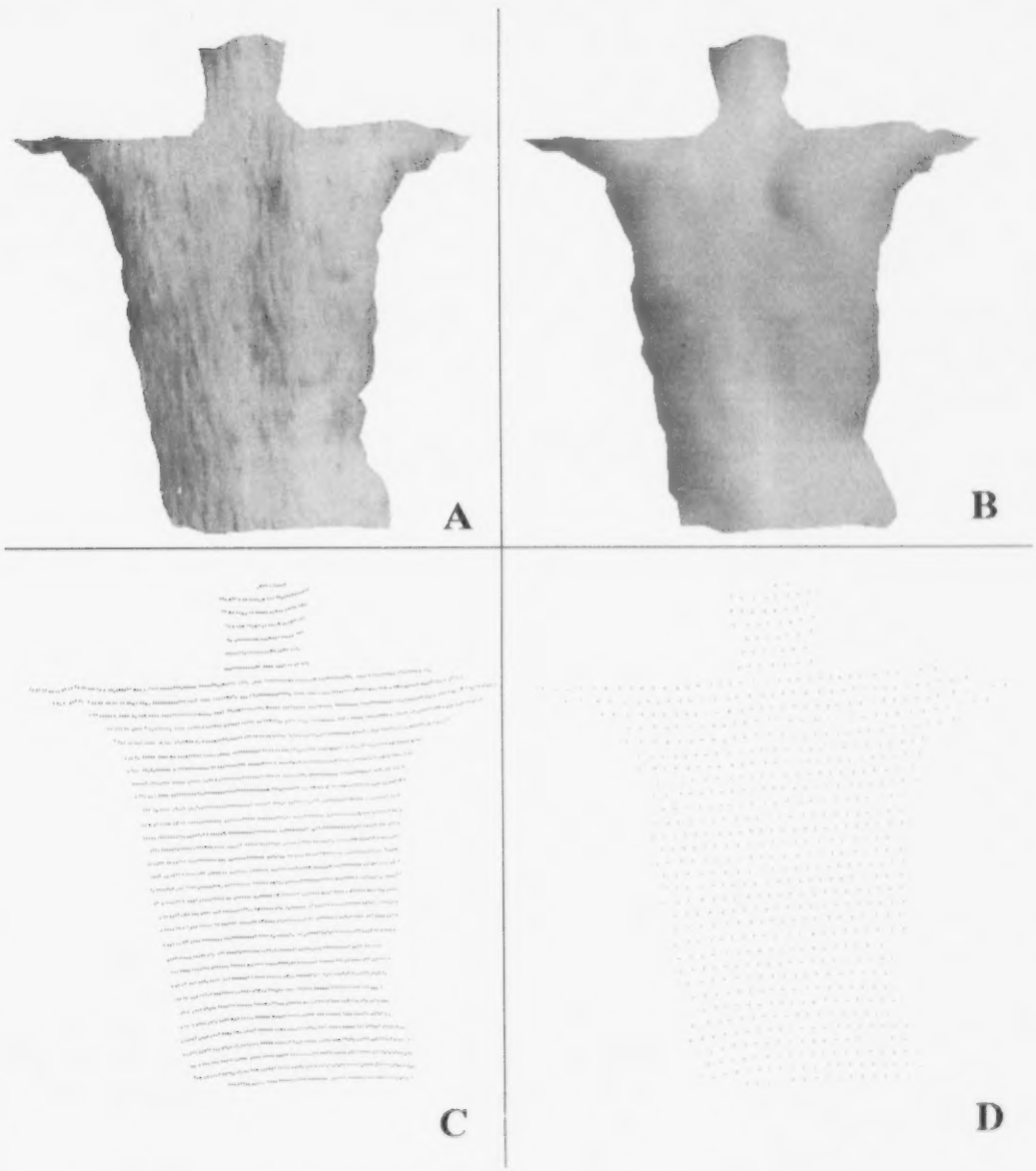

Figure 2.6: Thin triangles cause noisy light reflections (Rasterstereography (line pattern) data):A, C) Gouraud shaded surface and data points with all thin triangles; B,D) Same area with close to equilateral triangles. Every fourth data point is used from the original data set.

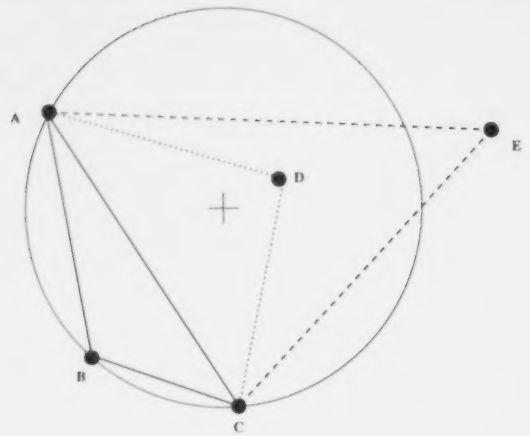

Figure 2.7: Delaunay circle

synonymous with maximizing the minimum interior angle.

One remaining challenge in defining which points constitute a valid triangle is incorporating the boundary. Boundaries have been successfully incorporated in the triangulation of regularly sampled volume data [19] containing white noise. The density of the sampled points is used as a guide to the location of the bounding points [28]. By adding a constant noise tolerance factor to the calculated density the boundary lines are made flexible enough to create a reasonable boundary. To elaborate: the tolerance level adds to the maximum allowable distance between data points which are to be connected.

Two methods of possible significance in three dimensional surface triangulation with boundaries are trimming the Delaunay triangulation and a triangulation based on point density. While trimming the Delaunay triangulation provided some useful basic concepts the core of the implementation will be derived from the density based triangulation.

2.2.1 Trimming the Delaunay Triangulation

An approach suggested by Veltkamp [26] is to define the Delaunay Triangulation and then trim away the invalid triangles to form a valid boundary. Since the Delaunay triangulation makes all connections to points in closest proximity [1], all valid connections will exist in its creation. However, the extraneous connections need to be removed. Using Veltkamp's own, newly developed graph theory each triangle is given various measures of goodness. Goodness is based on a combination of the ratio of triangle side lengths, and interior angle proportions. Those triangles with large angles facing the initial outer boundary (obtuse triangles) have low goodness factors and become candidates for removal.

It is easily seen that one could remove all inappropriate triangles to reveal the true boundary of figure 2.2. There are no points in the interior of the vase of figure 2.2. It is not clear whether only the appropriate triangles will be removed in the case where some data points are not a part of the boundary. The criterion that Veltkamp suggests for triangle removal are based on the knowledge that there are no interior points. For this reason, the immediate literal use of Veltkamp's methods would not be appropriate for our data.

2.2.2 Point Density based Triangulation

The theoretical algorithm developed by Hoppe [20] uses an estimation of the data density to define both the surface boundary and the valid triangle connections. Unlike the graph theory presented in Veltkamp, Hoppe et al. [20] take advantage of some

characteristics of the data sets. It is clearly stated that, "it is also impossible to recover features ... in regions where insufficient sampling has occurred" [20]. The density of the points must be estimated. A circle, which is guaranteed to hold at least one data point, has a diameter, ρ; where ρ is proportional to the density. Taking into consideration noise ϵ, a maximal distance between points is given as $(\rho + \epsilon)$.

Tangent planes are first defined for every point in the data set. This is accomplished by defining a neighbourhood size in terms of points. The user would pick an acceptable number, and the tangent plane would be a least squares fit of the neighbourhood to a plane. An automated technique to calculate the neighbourhood size was recommended by Hoppe [20], but had not yet been implemented. Hoppe [20] also suggested that in range images (depth maps) there may be a means of using information from the data capturing technique to define a neighbourhood size. The tangent planes create a locally linear approximation to the surface.

A Euclidean Minimum Spanning Tree (EMST) was constructed in $O(n^2)$ which connects the centers of all tangent planes into a graph. This graph was used to eventually connect all points that are in closest proximity. A modified marching cubes [7][24][20] algorithm is used to subdivide the area into cubes slightly smaller than $(\rho + \epsilon)$. Any tangent plane cutting through such a cube constitutes a possible border plane. The point nearest this plane is then connected to the nearest points in the neighbouring cubes to form the surfacing triangles.

A triangle optimization follows, which collapses edges that may have been falsely formed. The criterion used to eliminate these triangles is to minimize the product

(*maximum edge length* × *circumscribing circle radius*). Within the article by Hoppe [20] there was no discussion on how well this criterion worked [20].

It would be possible to directly implement their algorithms, but Hoppe [20] points out that the algorithm needs improvement before it can be automated. The density of the points needs to be known before the process begins so an automated means to calculate the minimum point density is required. The neighbourhood size (related to the density) is an input parameter that should be derived in an automatic routine. The speed and accuracy of the algorithm depends on these inputs so they become a crucial part of the triangulation.

The truly relevant portions of Hoppe's [20] procedures are those which can be translated into two dimensions. The calculation of point density and maximal connection line length can be translated into two dimensions directly. The marching cubes algorithm could be translated into a marching rectangle algorithm. However, there is no means of translating the tangent plane idea to a tangent line criterion that would still make sense for the data of interest. As with Veltkamp's [26] theories, the data used in [20] was all strictly boundary data. There is never any non-boundary or interior points. If there were no interior points in the trunk images, the theories presented in Hoppe [20] would be directly transferable. The existence of the interior points requires a slightly different approach to defining the boundary of our data sets. The methods used are presented in the chapter on the triangulation implementation. The two methods previously described both contributed to the form of the algorithm. The basic concepts of triangulating based on a point density, and removing trian-

gles after the triangulation to optimize the surface, are unique to the two methods described above and have directly lead to the implementation described in chapter 4.

2.3 Noise Reduction

When developing any system, maintaining accuracy is of some importance. Assuming that a triangulated description of the surface can be obtained, and that the surface lies strictly within the boundary of the data, the surface will still be inaccurate due to the noise in the data. In the research preceding this work [28], surfaces were produced that showed the potential problems of leaving the noise as part of the data set. Noise can result in ripples or a wavy appearance to the whole surface, causing the surface to appear unrealistic. A more serious side effect of the presence of noise is it can mask true features or produce a surface that appears to have features where none actually exist. With monitoring for scoliosis, the critical point is whether there has been change in the trunk. Noise can either hide the change or give the false impression that change has occurred. This can greatly hamper the usefulness of the trunk measures. These false or phantom features could show up as a bump on the back surface. When making surface comparisons between checkups on the same patient those phantom features will generally not appear in the same positions thus making characteristics of the surface inconsistent. Misleading visual cues such as phantom features need to be removed to make the surface data useful for analysis.

A smooth surface is not necessarily ideal. Smoothing can produce the same undesirable effects as noise. It can hide features. It is not suggested that the sampling

will pick up all features no matter how small, but the data set contains information of features of a specific dimension/frequency. If those size features were to be smoothed over, the resultant surface, regardless of its visual appeal is no longer as useful and is potentially misleading.

Splines [18] are smoothing curves which interpolate between points and pass through the actual data points. Splines in the presence of noise will create fluctuations in the curve. Although the surface would appear smoother than the flat triangle connection of the original surface it requires both that noise be already minimized and that some organization of points is known. Neither of these conditions hold for our data, which makes splines an inappropriate choice for surface generation.

In the world of data visualization for volume or three dimensional data, there is a lack of published material on removing the effects of noise. Noise is recognized to exist in the capturing of surface data [20][12], but it is only significant in structured light imaging [12]. Many of the techniques for data capture such as magnetic resonance imaging and laser range scanning have considerably smaller noise levels, due to the precision in the equipment being used. However, these methods of data capture are not yet appropriate for back surface data. The patient needs to be lying for a magnetic resonance image. Lying down causes the back surface to change which will mask much of the surface asymmetry seen in the standing position. The long scanning times required for laser scanning would require that the patient not move or change posture for many seconds. Since patients constantly move due to sway and breathing, laser range scanners would contain motion artifacts.

2.3.1 Noise Reduction through Equation Fitting

The research done by W. Frobin and E. Heirholzer [12] examines interpolation techniques to fit structured light data to a regular grid and presents measures of accuracy. The purpose of their study was to produce a data set that can be used in surface shape analysis. The method presented [12] was a smoothing algorithm that takes a neighbourhood of points and least square fits the data to a second order equation. Noise reduction was a side effect of fitting the polynomial to a relatively large area. The size of the neighbourhood of influence was a parameter that is empirically defined so that it works well with their surfaces.

However, random distributions of points require lengthy matrix inversion calculations when fitting the points to a second order patch. To eliminate this problem a linear interpolation was performed between points in a row and again between rows of points. This interpolation creates a grid on which the new depth (Z) values will be placed for the smoothed surface. The ability to do this implies that there needs to be a priori knowledge of the organization of the points, as there is with the structured light data they were using (figure 1.4 and 2.6 C) [18]. High densities of points lie in the rows while the distance between rows was less dense. For our data sets, the initial interpolation to a grid is not possible using the same methods since implicit neighbourhood information is not readily available. The data sets which will be used will vary depending on the periodic pattern of light projected onto the back surface of the patient. Both the line rasterstereography and the dot projection patterns need to be processed using the same algorithms.

2.3.2 Resampling to Organized Points

Without a sufficient method to reduce the effects of noise, the data would require resampling so that known techniques can be used. As was suggested by Frobin [12], Tiede [24], and Turk [25] a set of data points can be linearly interpolated to a regular grid. Unorganized points can be linearly interpolated if a triangle mesh is defined for the surface [25]. A regular grid can then be produced by choosing the surface points such that they land on grid intersections. The accuracy of the resampled points was not a concern in [25], but will be discussed here in the noise reduction chapter.

2.3.3 Filtering

Assuming that a regular grid of points is now at our disposal, there are multiple techniques for minimizing the effects of different types of noise. It was not known at the onset of this study the actual types of noise that were present in the data or how it was distributed. However, both linear and non-linear filters are useful for removing the effects of noise in two dimensional image processing [10]. By treating the three dimensional surface data set like a two dimensional intensity (depth) image, it was expected that image processing filters could be applied successfully.

The local average filter (linear) is defined in a kernel which is convolved with a regular grid (depth map) to remove white noise. High frequency components are assumed to be noise and they are minimized by this filter. The size of kernel is determined by the desired region of influence. Another linear filter is the ideal filter (sinc function) which acts similarly, as a low pass filter at a specific frequency. The

median filter (non-linear) removes spike noise from a regular grid while retaining slope and edge information. Each filter has its advantages and disadvantages. The local average filter blurs edges (ridges on the surface), the median filter only removes spike noise , and the ideal filter cannot be physically implemented: only approximations to it.

Each filter was tested. Another filter will be derived from these standard image processing filters to produce the desired noise reduction for the back surface data. All these methods are detailed in chapter 5 on the implementation of noise reduction methods.

2.3.4 Summary

Based on surface reconstruction literature a proposed method of creating a realistic surface is now developed. That surface reconstruction will involve the description of the data set in terms of a triangle mesh, the subsequent determination of the boundary of the data set, and the reduction in the effects caused by noise in the data through filtering. The next three chapters deal with each process separately starting with the boundary generation problem.

Chapter 3

Boundary Formation

3.1 Background

In all boundary or sectioning schemes [17][12][14] with the only found exception being [26], there is a manual decision making process which determined an initial boundary or sections to be made.

Veltkamp [26] comes the closest to finding an automated, rigorous method of generating a boundary from random points. His research was looking for the best theoretical surface for any irregularly sampled object. Although Veltkamp states that no assumptions can be made about his data sets, our data allows and relies on the assumption that the surface is adequately and relatively uniformly sampled. These two conditions must hold to recover the surface features of interest.

For any data capturing technique, in which the data points themselves are the only information available, the computer boundary generation should be able to find the boundary with as much certainty as a manual generation. This assumption rules

out any necessity for manual intervention in the boundary generation process. Any surface property/characteristic which requires manual insertion is not part of the data set and indicates that sampling is not adequate to capture such features. Trying to automate the insertion of boundary characteristics such as breaks or sharp cuts which cannot be captured by the data acquisition system was not considered.

The classification of surface topologies for a triangle mesh [27] (figure 3.1) suggested that, by using only the information inherent in the data capturing technique, an efficient and unambiguous boundary can be generated from the back surface data. These classifications are determined after generating the triangle mesh, which suggests that the boundary should be inherent in the data. By examining the data there should be a best guess for where the data begins and ends, and that is where the boundary should be defined. Any and all features of the surface should be present in the data and be defined by the data. The most important class for boundary detection was the boundary class. After a proper triangulation, the boundary is automatically defined by the data points which are not completely surrounded by surfacing triangles. Working with this realization, the next step was to find the appropriate triangulation which would automatically triangulate within the natural boundary of the data.

3.2 Implementation

Rather than explicitly defining a boundary, the problem was approached as defining the longest connection line in the connectivity graph of the data. The triangulation routine would not be constrained by a boundary, but rather would be constrained

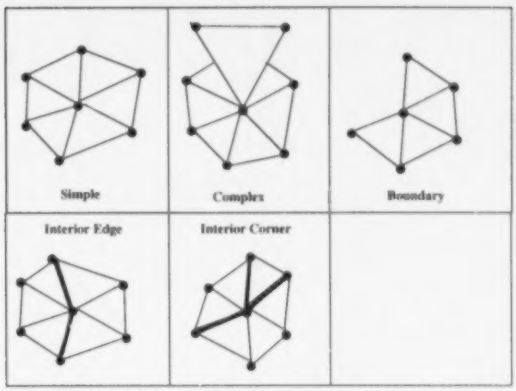

Figure 3.1: Point Classification for a triangulated mesh

by the length of the triangle edges. In three dimensions, this has the effect of not connecting points on the surface which are across the back, because they are farther apart than allowed. Since the triangulation will be accomplished in two dimensions, the boundary that will be created by such a triangulation will connect all points on the boundary which are, at most, some pre-set distance apart. If the density of points is relatively uniform, all boundary points will be approximately the same distance apart. If this distance is used to limit the line length, a completely connected surface can be derived which is enclosed by the appropriate boundary.

It was desirable to automate the line length threshold calculation. A method was developed to find an approximation to the point density in two dimensions. From the density, the maximum connecting line length is calculated.

3.2.1 Density

The list of XYZ coordinates were back projected onto the X-Y plane. The result of the back projection is a distribution of points in two dimensions. The center coordinate

in the X and Y directions is used as an approximation to the centroid of the data points. The chosen centroid will be well within the boundary of the data sets of interest, because the imaged area of the back is approximately rectangular in shape. To obtain an overall approximation to the density, a density value is assigned to each point in the data set. The most common value (mode) of these densities is used as the density approximation for the whole surface.

To calculate the density for each point, the points are sorted in increasing distance from the center of the bounding rectangle. Then, a density is assigned to each point as follows:

$$\rho(\mathbf{n}) = \frac{\mathbf{N}}{\pi \times \mathbf{Radius(n)}^2} \tag{3.1}$$

where:

ρ(n) is the density as calculated for point n,

N is the number of points within a radius of Radius(n) from the centroid, and

Radius(n) is the radial distance from point n to the centroid of the data set.

The densities will be more inaccurate as the circular area becomes larger than the actual boundaries of the data. The densities which are calculated for areas containing very few points will also be prone to errors, because a large area is required over which an average density can be calculated. A sample list of densities (figure 3.2) for each

point in a surface demonstrates how the density numbers vary as the area increases, over which the approximation is taken. The data is not perfectly uniform so when few data points are considered the density will not be a reliable average which can be generalized to the rest of the surface.

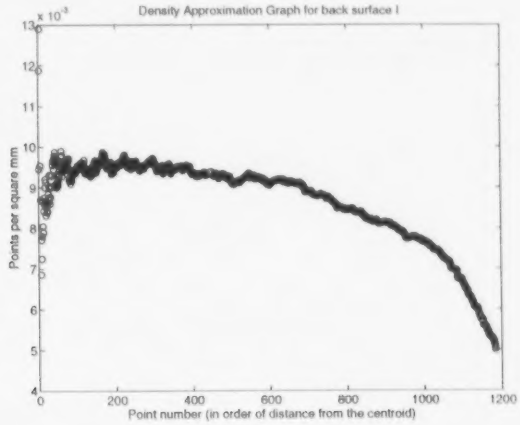

Figure 3.2: Point Number vs Density Calculation

The most common density, or mode, is taken to be a good approximation to the data sampling density. A histogram with 256 vertical lines (divisions) (figure 3.3) shows where the mode is located: at $\approx 9.5 \times 10^{-3} Points/mm^2$. Rho ($\rho$) will represent this theoretical density of points from here on and is approximately 1 $Point/cm^2$ for all surfaces used as examples.

3.3 Maximum Line Length

The square root of the density is the theoretical maximum distance between any two points in a uniform point distribution. Since the point density is not uniform throughout the data set, a correction (safety) factor is introduced to ensure all points

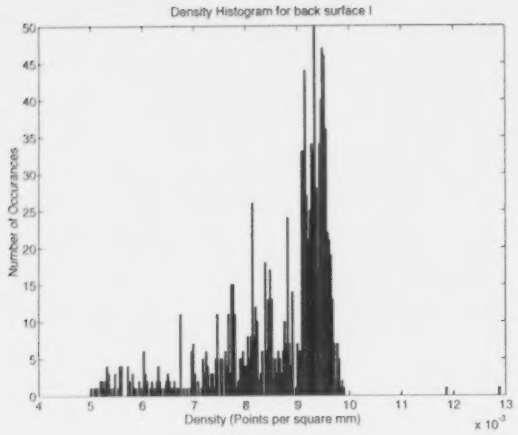

Figure 3.3: Density Histogram

will be connected which reside within some maximum distance from one another. The non-uniform stretching or shrinking of the light pattern in the Y (vertical) direction varies depending on the relative angles between the projected light and the camera's focal direction. The current data acquisition system (figure 1.2) has a projector angle of 30° from the horizontal and the camera is parallel to the horizontal. The maximum connection line length required to completely connect surfaces at various angles is given in figure 3.4. For line rasterstereography the point density is different, and for camera setups which change the distance of the projector camera or the density of the projected patterns the density calculation will reflect those differences. For a single consistent camera setup the same approximate density will always be calculated, but since there are many varying possible data sets for back surface reconstruction the density must be calculated without the base knowledge of which camera setup was used.

If the surface is inclined at 30° or less there is no data acquired, because the

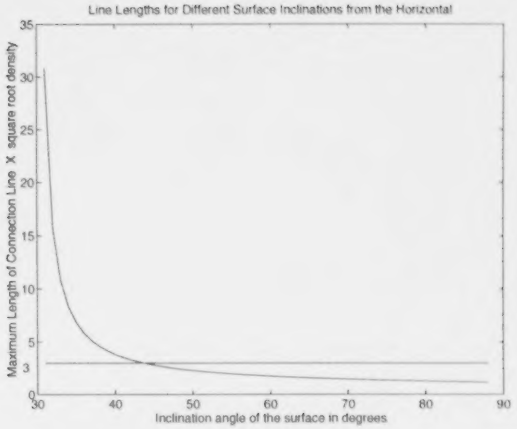

Figure 3.4: Connection lengths for surfaces with different orientations

projected light will no longer reflect off the surface towards the camera. The same is true when the surface is inclined at 180° (laying flat). The fact that most back surfaces approximate a vertical plane when standing, makes structured light an appropriate imaging technique. Only features such as protruding shoulder blades and waist creases cannot be captured effectively using this light projection technique. If a back surface were to be so severely deformed as to have large areas occluded by features on the back, this data acquisition system would no longer be appropriate to image the back surface. The density of points on the back surface would no longer have implicit neighbourhood information provided by a relatively uniform density of points. In the presence of occluded or sparsely sampled areas, the density of points would no longer be adequate to either describe the surface below in enough detail for analysis, or to make decisions as to how to connect the data points into a single solid surface.

When choosing the correction factor for the maximum line length the range of back inclination angles must be encompassed which will include all back surfaces of

interest. The maximum line length was chosen to be able to connect all surfaces features at angles greater than 45°. The extreme case would be a whole flat surface at 45°. In figure 3.4 the horizontal line shows that in order to connect all points in a planar surface at 45° the connection lines would need to be at least three times the theoretical maximum line length. It has been empirically determined that a line length of $(2.5 \times \sqrt{\rho})$ is sufficient to completely connect all back surfaces in the current structured light data sets. Data sets from patients with severe back deformities could be completely connected with a line length $(2.5 \times \sqrt{\rho})$. This tested the extreme cases where the density of the data points would have the least uniform density. The density is the least uniform because the angles of the back surface go through the largest fluctuations thereby altering the displacement in the Y coordinate by the greatest amounts.

However, data sets which are not derived from structured light data acquisition systems, such as hand digitized data, may not have the inherent relatively uniform density. In such a case, the multiplying factor may need to be increased to fully connect the entire data set. Increasing the multiplying factor affects the accuracy of the boundary, just as with any other surface, but it would be necessary if the entire data set was to be fully connected.

By changing the multiplying factor, the length of the allowable connection line changes. By increasing the factor, the allowed connections become longer and the likelihood of making a invalid connection outside the region of the back surface increases. If the multiplying factor is too small then all valid connections cannot be

made, leaving holes in the surface or not including data points in the surface.

3.4 Further Considerations

In surfaces with points missing (features that were occluded, or errors), or when using random light patterns, a larger multiplying factor is required to completely connect the surface. Otherwise, the surface triangulation routine will leave holes in the surface wherever the necessary line length required to span the hole is too long. The problem with choosing too large a correction value is that all density information becomes irrelevant, and line lengths become too long. The smallest correction factor possible is desired so that areas outside the surface are not included in the surface.

Another solution to finding an accurate point density is to rectify the data. This would rotate all points into the plane of the projector and regain the regular grid pattern that was projected onto the surface. Rectification is accomplished by multiplying each point by a rotation matrix. The rotation matrix would depend on the angle between the projector and the camera, thus it requires that more information about the data be known. By multiplying the vertices by the matrix all points will once again be in the periodic light pattern which was projected on the back surface. All rotation should occur about the focal point of the projector (X,Y,Z = 0,0,0). This ensures that the focal point of the projector is the same before and after rectification and keeps the data points centered. If the data is not already centered about the focal point of the camera a translation is required before rotation can be applied. Translation to the focal origin and rotation can be applied consecutively in a matrix

multiplication with the data points to apply the desired rectification. The necessary multiplication is seen in equation 3.3

Assuming that the periodic pattern is of the necessary uniform density, the density approximation calculation and triangulation process will be faster and will work well for more types of surfaces. An example of a data set rotated into the plane of the projector is seen in figure 3.5. The only problem is that the data acquisition system needs to be structured light and the projector angle must be known. This requires more a priori knowledge and places some restrictions on the data used.

$$[V'] = [Rotation][Translation][V] \tag{3.2}$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & sin\theta & 0 \\ 0 & -sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & X_\circ \\ 0 & 1 & 0 & Y_\circ \\ 0 & 0 & 1 & Z_\circ \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.3}$$

3.4.1 Summary

The process to find the length of the longest valid connection between data points has been generalized to any data distribution with a relatively uniform density. By minimizing this length, the boundary is kept as close to the actual boundary as possible. The accuracy of the boundary is maximized by limiting the connection of points to those within the maximum line length. This length will be used to ensure that connections in the triangle mesh generation (next chapter) will not go outside

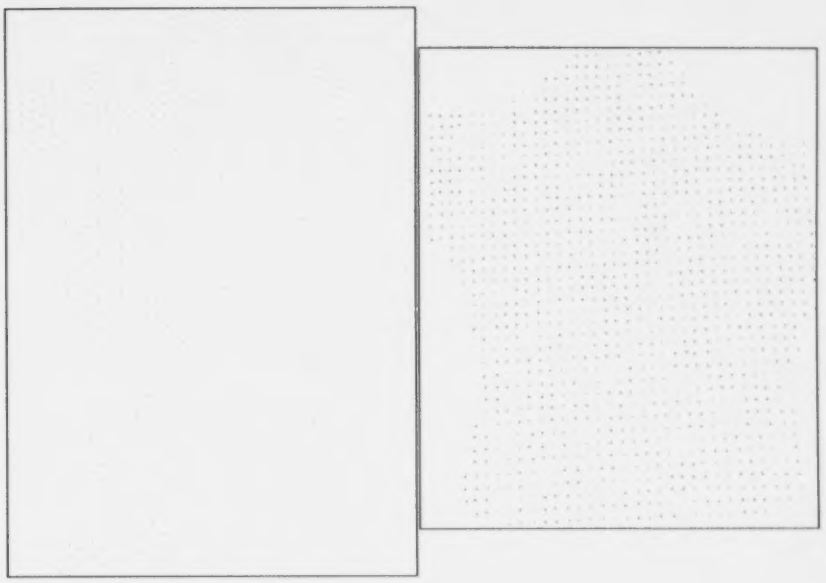

Figure 3.5: Left: data distribution before rectification; Right: rectified data

the boundaries of the data sets.

Chapter 4

Triangulation

The information sources available for the triangulation process were:

1. The automated density calculation described in the previous section.

2. The data acquisition method cannot capture folds in the surface.

3. A list of points in their Cartesian coordinates in the format:

X1	Y1	Z1
X2	Y2	Z2
.		
.		
.		
Xn	Yn	Zn

Xn, Yn and Zn are all numeric (integer, float, or double)

The criterion for the final surface representation which are relevant to the triangulation process are:

3) That the surface lies strictly within the boundaries of the data

4) That a fully connected solid surface be created

To insure the connection of all points in the surface, the connections were made to every point within the maximum line length. Triangles were made out of three such connected lines. By stating that strictly no connecting line length should exceed the maximum line length, it was postulated that the boundary will be implicitly constructed and the surfacing triangles would all lie within its bounds.

The triangulation process is divided into two stages. Connecting the points into triangles is the first stage of the triangulation. The second stage addresses the concern for optimal triangle shape. There are different criterion for deciding the best proportions for a triangle. It was decided that the Delaunay triangulation would be the best, because it is a unique triangulation and connects the most local three points in all cases. Locality was described in figure 2.7 by a circular area that contains no data point in its interior and circumscribes the three points which define a triangle.

The two stages will be discussed separately, because that is how they were implemented. First a triangulated mesh is created and then a re-triangulation is performed to optimize the triangulation. Re-triangulation (different versions of re-triangulations can be found in [25], [15], [20], and [24]) has been demonstrated to successfully adjust triangle shape after a mesh has been created. The same concept is used here so that the initial mesh generation can be more quickly defined.

The optimization stage takes a fraction of the time for mesh generation, but would take more time if done in conjunction with the mesh generation. It is more efficient

to do the steps separately, rather than together, because to optimize a triangle the nearest neighbours to that point need to be known. The initial mesh provides knowledge of the nearest triangles to a given point and the searching time is subsequently reduced. This may not have been an actual time saving splitting of tasks had there not been an efficient means to defining the initial mesh without optimization. This efficient process of creating the initial mesh now described in detail.

4.1 Initial Triangulation (Stage 1)

As stated, all points and only those points within a distance less than or equal to the maximum line length (D) were connected. This allows the two dimensional plane to be subdivided, thus subdividing the task of searching for nearest neighbours.

The total data set is divided into a regular square grid in the XY plane. The dimension of each square of the grid is equal to the maximum line length. The process of triangulation takes place within a neighbourhood of four such grid squares. Four adjacent squares (figure 4.1) are chosen and an exhaustive nearest neighbour list is constructed for each point within its interior. Only those points within the four adjacent squares are candidates for connections. Only those points within a Euclidean distance less than the maximum line length are added to any other point's list. Those lists are then sorted in order of increasing distance from the original point. This is necessary to ensure that no triangles can possibly overlap.

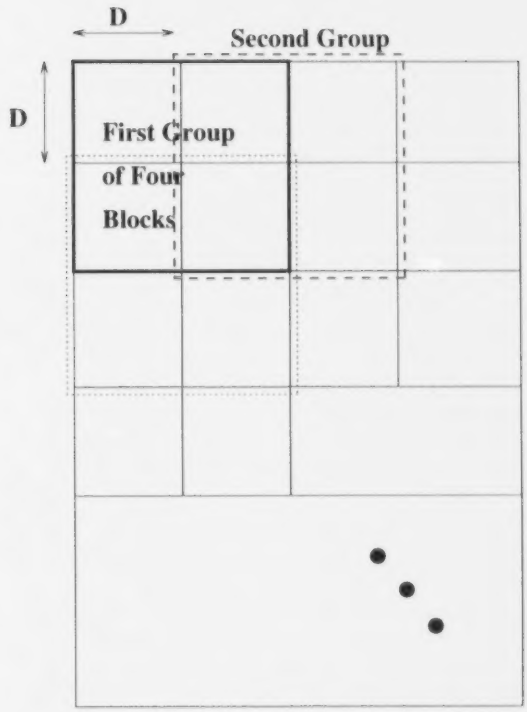

Figure 4.1: Triangulation Partitioning: The width and height of the smallest squares are equal to the maximum allowed connection line length

4.1.1 Triangulation about one Point

Each point within the group of four grid squares is iterated through one at a time until it is either surrounded by triangles or has run out of neighbours in its list. The steps involved in choosing and storing the information for each triangle is best explained using an example. The sample neighbourhood is given in figure 4.2. The sorted lists of nearest neighbours is given in table 4.1. One point is chosen (in no particular order) and is triangulated.

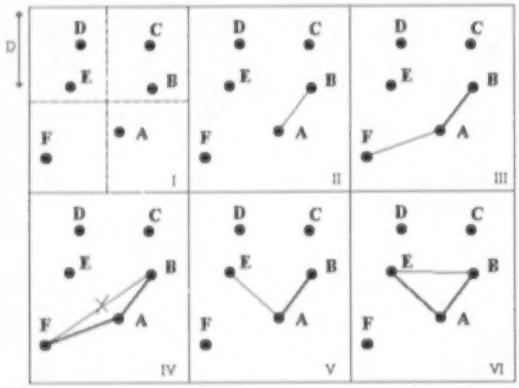

Figure 4.2: Sample Point Neighbourhood: Steps taken in triangulation

Point	A	B	C	D	E	F
Valid Distances	B F E C	C A E D	B D E A	E C B	D A B F C	A E
Invalid	D	F	F	A F		B D C

Table 4.1: Creating one Triangle: List of Nearest Neighbours, Those points below the dividing line are beyond the longest acceptable connecting line

Point A will be chosen as the first point for all triangles and the point nearest to it (first in A's list of neighbours) is taken as a candidate for the second point in a triangle. Line A-B would be one edge of a potential triangle. If there can be found a point other than A or B that is in both neighbourhood lists of A and B it becomes the third point in the triangle. Since E is the first point found in both the lists of A and B it becomes the third point (NOTE: A's list is always traversed first when looking for a candidate point).

Now that a potential triangle has been found it must be determined that it does not overlap any portion of any triangle that has previously been defined. First, the storage structure of a triangle must be known. A structure had been previously created which has space to store triangles for every point in the data set. Each point gets a copy of the triangle information:

Example storage for point E:

- the identifiers of the other two points (A, B)
- the respective angles the connecting lines (originating at the original point) make with the horizontal ([A-E 300°], [B-E 2°])

The points are stored in order such that if the two connection lines originate at E, by pivoting the first line counter-clockwise towards the second line, one of the interior angles of the triangle will have been swept through (figure 4.3).

To determine whether a triangle overlaps another, only the three points in the new triangle need to be examined. All triangles defined in the triangle storage structure

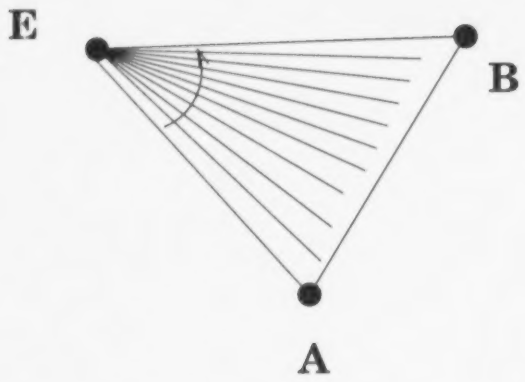

Figure 4.3: Triangle Interior Angles: Line AE will be stored first because it sweeps counter clockwise to line B-E through the interior of the triangle.

under any of these points are examined further (in this case (A,B and E). The interior triangle angles of the new triangle must strictly not include any portion of an interior angle of another, previously created triangle.

The determination of overlap is best described in the following pseudocode.

```
/* TEST FOR OVERLAP */


if ( new_triangle and old_triangle share a vertex (V) ) then {

   /* four triangle edge lines will originate from V */


   minimum_angle = angle(V,old_triangle.edge_1);

   maximum_angle = angle(V,old_triangle.edge_2);


   first_new_angle = angle(V,new_triangle.edge1);

   second_new_angle = angle(V,new_triangle.edge2);
```

```
   if ( first_new_angle is between
      maximum_angle and (minimum_angle + 180 degrees) ) then
      The triangles do not overlap;
   elseif (second_new_angle < 360 degrees) then
      The triangles do not overlap;
   else
      The triangles overlap;
}

if ( The triangles do not overlap ) then {
    ADD_TO_VALID_TRIANGLES (new_triangle);
}

/* TEST FOR COMPLETENESS */
if ( unused(vertex(i)) ) then {

   if (sum of surrounding triangles of vertex(i) == 360) then
      SURROUNDED = TRUE;
      /* go on to the next vertex */
      i = i + 1;
}
```

```
else /* continue */

      triangulate (vertex(i));


}
```

TEST FOR OVERLAP

A description of the above pseudo-code explains the process. Suppose there are two triangles that share one vertex (V), then there will be two new triangle edges and two old triangle edges that are coincident at vertex V. To determine whether the two triangles overlap a new frame of reference is created. The first line in the new triangle structure (the first line encountered traversing the triangle in a counter-clockwise fashion), is given the angle 0°. The remaining three lines are made relative to this one. If the angle the old triangle's first edge lies between the last edge of the new triangle and 180° the triangles do not overlap. If the previous condition was not true, then if the last edge of the old triangle is less than 360° then the triangles do not overlap, otherwise the triangles do overlap. If no triangles with coincident vertices to the new triangle overlap, then no triangles overlap and the triangle is added to the list of triangles.

COMPLETENESS

The next step is the same for whether the triangles overlap, or if a new triangle was added. If the current point being triangulated (A) is surrounded by triangles: the sum of all interior angles of all triangles originating at A equals 360°, the next point

in the region is examined in the same fashion. If point A is not surrounded, line A-B is used again and another third point is sought. Once two triangles containing line A-B have been added, the next point in A's neighbour list is chosen as a candidate for connection.

One point is triangulated until it has no remaining neighbors which are candidates for connection, or the point is completely surrounded by triangles. Each point in the group of four grid squares is triangulated followed by the next group of four. The order in which the groups are chosen is irrelevant, so any systematic progression through the entire area will suffice.

4.2 Triangle Optimization (Stage 2)

Retriangulating a triangle mesh is used when an initial mesh is not the best representation for a surface. It can be used to reduce the number of triangles [23][27][25], to improve the shape of the triangles [15][20], and to varying the level of detail in a representation [23][25]. The retriangulation will, in this case, optimize the triangle shape as defined by the Delaunay criteria. To reiterate: the triangle is optimal if a circle circumscribing the three vertices strictly contains no other point in its interior.

The method used to optimize the triangles is based on an equivalent set of criterion to the Delaunay criteria for optimal triangles. Sibson [14] illustrated the optimization with two adjacent triangles incident on one edge (figure 2.7). Adjacent triangles refer to triangles neighbouring on the edge opposite both triangles' largest interior angle. It is proven in [22] that the alternate triangles are chosen only when a point is in the

interior of the circle circumscribing the points of one triangle.

Only those triangles in the mesh with an interior angle greater than 90° will be examined for retriangulation. It is proven below that by only examining these triangles all potential optimization will be done.

Statement : At least one of the two adjacent triangles requiring optimization has an angle greater than 90°.

PROOF of the above statement (by contradiction): It has already been established that a circle circumscribing the vertices of one triangle has the fourth point in its interior (thus the need for optimization), and that a fully connected non-overlapping triangle mesh is the origin of the triangles.

Premise:

A) Triangles do not overlap one another.

B) There is a non-optimal triangle adjacent
to another triangle along their longest edges.

Assumption: There exists two acute triangles adjacent on one edge which lie on or within a circle circumscribing one of the triangles' vertices.

1. A right angle triangle has its longest edge equal to the diameter of a circle circumscribing its three vertices.

2. Any acute triangle will always contain within its interior the center of the circle circumscribing its three vertices.

3. Any obtuse triangle will strictly never contain within its interior the center of the circle circumscribing its three vertices.

4. There is only one center to a circle and it cannot be included in the interiors of two triangles which do not overlap.

   Since, by premise A, the triangles do not overlap, statements 2 along with 4 directly contradicts the assumption. Thus, by contradiction, at least one of the two adjacent triangles of a potential optimization has an angle greater than 90°.

Only obtuse triangles are singled out for the optimization stage. It is a simple matter to find the acute or obtuse triangle that is adjacent to the long edge of the obtuse triangle. All triangle information that is stored in the initial stage can be used to find the adjacent triangles to each obtuse one. Thus, the method for finding the candidates for optimization has been established, and it remains to be shown how the optimization is achieved.

When the two triangles of an optimization have been located, it is then determined whether the fourth vertex lies within a circle circumscribing the vertices of one of the

two triangles. If it does lie within the circle, the quadrilateral created by the four points is divided along its opposite diagonal: creating two new triangles. By choosing the alternate diagonal it is shown by Sibson [22] that the points no longer exist within the other's circumscribing circle. One of the two new triangles may still be obtuse, in which case it will be added to the list of triangles yet to be optimized. This process converges as the number of obtuse triangles decreases and the circumscribing circles become smaller and smaller until the optimal triangulation is complete.

The final results of the triangulation are written to a file. The description of the triangle mesh is given in terms of vertex triplets which are integer indices into the array of three dimensional points: as in the following table.

0	1	2
14	9	12
.		
.		
.		
Point1	Point2	Point3

4.2.1 Summary

This constitutes the last measures that were taken to satisfy the connectivity of the surface and the boundary generation. The next processes will be aimed at transforming this connected surface into a noise reduced and thus more realistic surface.

Chapter 5

Noise Reduction

The local average filter, median filter and one custom filter was used to minimize noise in the data sets. For the standard filters (local average, and median) to be applied it was necessary to resample the data into a regular grid so that convolution with a kernel was possible.

5.1 Resampling

Resampling took the form of a bilinear interpolation between points in their respective local neighbourhoods. It was accomplished by taking the output from the triangulation and choosing points which were regularly spaced in the X-Y plane and residing on the planar triangle surfaces in the Z coordinate.

The resampling introduced an error in the data which was proportional to the sampling interval. The maximum possible error is equal to half the resampling interval for bilinear sampling. This is the maximum distance that a resampled surface will be above or below the original data points. This assumes that the smallest surface feature

is a half sinusoid or half circle with a diameter equal to twice the original sampling frequency. In order to maintain the same feature size a resampling frequency of twice the original is necessary (by Nyquist's sampling theorem). Since accuracy was one of the criterion for the final surface representation, the resampling frequency needs to be such that the maximum possible error in the Z coordinate does not exceed the original data acquisition uncertainty. However, the higher the sampling rate the more triangles that need to be rendered on the screen in the final representation. The goal is to find a balance between the number of triangles and the error introduced due to resampling.

The sampling interval was chosen so that the on screen rendering took less than a second for a surface with approximately 2000 original points (Dependant on the computer hardware). The smallest sampling interval that complies with this criterion is one third of the original sample spacing. When sampling at this interval the maximum possible height difference between the original surface and resampled surface is $1.5mm$. This is the vertical (Z) displacement of the resampled surface from the original data points.

The inputs to the resampling routine are the uncertainty value for the data acquisition, and the original density estimate as calculated during the triangulation process. The surface is then resampled with the restriction that the original uncertainty is not exceeded. In the worst case the resampling and noise reduction processes will move any point a distance in the Z coordinate equal to the original uncertainty. This is analogous to finding a best fit curve through error bars for two dimensional

data sets. The error bars would be 2mm above and below each data point in this case and the surface will always pass between the upper and lower bounds of the error bars for each original data point.

> E.g. The original uncertainty is $2mm$ in the Z coordinate and resampling at one third the original sampling density would move the surface at most $1.5mm$ in the Z coordinate. The convolution with a noise reduction filter could move any point $0.5mm$ in the Z coordinate without the sum of the resampling error and noise reduction exceeding the original uncertainty.

5.2 Standard Filters

A local average filter and a median filter were used in noise reduction. The local average filter is meant to remove high frequency components in the data. It was implemented in a slightly modified form so that a tolerance level limits the change in the Z coordinate. This tolerance level is calculated using equation 5.1. Whenever the absolute change in depth is greater than the tolerance level the point is clamped at the tolerance level.

$$Tolerance = NoiseLevel - ResamplingError \tag{5.1}$$

The neighbourhood encompassed by the filter kernel must include the central point and its most immediate neighbours. Using a neighbourhood size that goes beyond these limits causes the highest frequency components in the surface (smallest

features) to be eroded (smoothed over). Using a smaller neighbourhood size reduces the influence of the neighbouring points and the filtering technique loses its ability to significantly change the values of the central data points, and therefore its ability to reduce the effects of noise. Smaller kernel sizes (i.e. 3×3 and 5×5) were applied to resampled data sets and the amount of noise reduction ranged from undetectable to insignificant.

A 7×7 neighbourhood in the resampled surface corresponds to an area on the original surface which contains approximately 9 points which is approximately a 3×3 neighbourhood of the original data points. If the original data was a regular grid (uniform density) then a 3×3 neighbourhood would be covered by a 7×7 kernel, but the non-uniformity in the back data requires a larger kernel. To ensure that close to 9 original data points are encompassed by the filter kernel, the minimum of a 9×9 neighbourhood is required for a kernel size.

5.2.1 Implementation

Both the median filter and the local average filter were tested on back surfaces using 9×9 kernels. The weighting given to each point in the kernels was set to unity. Unity weighting is the standard way to implement a local average or median filter. Other weighting combinations were experimented with, but no visual improvements were detected. Based on the results of the filtering with unity weighting, there was no reason to explore the use of other weighting factors.

The local average filter (kernel = 9×9 array of 1's) was convolved with the

resampled data and retriangulated for display. The triangulation routine used for the display of the resampled data is simplified because of the explicit knowledge of neighbourhood information within a regular grid. The central point is replaced with the average of all its neighbours in the 9×9 area. If the new value of the central point differs from the old value with a magnitude greater than the tolerance, then the new value is clamped at the tolerance level.

The median filter chooses the median value of a sorted list of points in an area under a 9×9 kernel. The points are sorted by their Z coordinates and the middle (median) value is replaced into the regular grid of points as the central value. As with the local average filter, the tolerance level was used to clamp the amount of change in a central value by the tolerance level.

5.3 Custom Filter

Using the knowledge that the points are distributed in a relatively uniform manner and that the triangulation of the points connects the points into local neighbourhoods, a custom filter was implemented. Each point in the triangulated mesh is surrounded by a finite number of neighbours. Those neighbours form a circle (or partial circle in the case of boundary points) about the center point (see figure 3.1). An average of the depths of all these points was thought to have a similar effect on the surface structure as the local average filter would. Resampling is not necessary for this method, so the error that would have been introduced from resampling is eliminated. This gives the filter more leeway to eliminate noise without exceeding the noise tolerance level

(equation 5.1). The tolerance level is still used to clamp the change in point heights, but that tolerance level is now larger.

5.3.1 Implementation

This method requires that the triangular mesh was previously created. From within the triangulation program, the depth values for each neighbour and the center point were summed and divided by the number of points in the summation. This average was then compared against the original depth value of the center point and clamped using the tolerance level. The resultant points were then stored and written to the triangle connections file as before. The triangle connections still remain valid after filtering, because the height (Z coordinate) is the only value changed for each point and the height was not taken into consideration during the triangulation.

Chapter 6

Results

Tests were performed to determine how closely the objectives were met by the chosen methods. The tests explored the degree of accuracy, speed of the processes, the level of automation, and the realistic qualities of the output for all processes. The best way to evaluate many of the results is through images of the process output. The images in this chapter are used to help illustrate the quality and accuracy of the different stages of the back surface creation.

6.1 Accuracy

Accuracy was evaluated differently for the three stages of surface generation: boundary detection, triangulation, and reduction of the effects of noise.

6.1.1 Accuracy - Boundary

For the boundary detection, the accuracy was determined by how closely the generated boundary was to the actual boundary of the patients back. This can be deter-

mined by overlaying the generated triangle mesh over a photograph of the patients back, which was taken at approximately the same time. In the example to be shown there was a couple of seconds delay between the taking of the greyscale photograph and the data set pictures. In figure 6.1 the computer generated boundary was shown as a black outline which stays within the physical boundaries of the patient's photograph. This showed that limiting of the line length eliminated the problem of long connections which create surfaces outside the physical boundary of the patients back. The boundary does not closely follow the actual boundary (there is a gap between the computer boundary and actual boundary), because the data itself was not acquired exactly on the actual boundary of the patients back.

6.1.2 Accuracy - Triangulation

The second stage of surface generation was to generate a triangle mesh. Although the boundary was created at the same time they both address different criteria for accuracy. The purpose of triangulation was to create a solid triangle mesh of the most local neighbourhoods of points. The triangles were to include all data points and were not to overlap. The accuracy of the triangle mesh can be seen as how closely the surface connections match the surface shape of the original surface. This was done in a two dimensional test which ensured that no holes were left in the surfaces and all data points were included in the mesh. Four examples of mesh creation can be found in figure 6.2. The first figure (I) shows a highly non-uniform distribution of points (not obtained using structured light techniques). This figure (I) illustrates

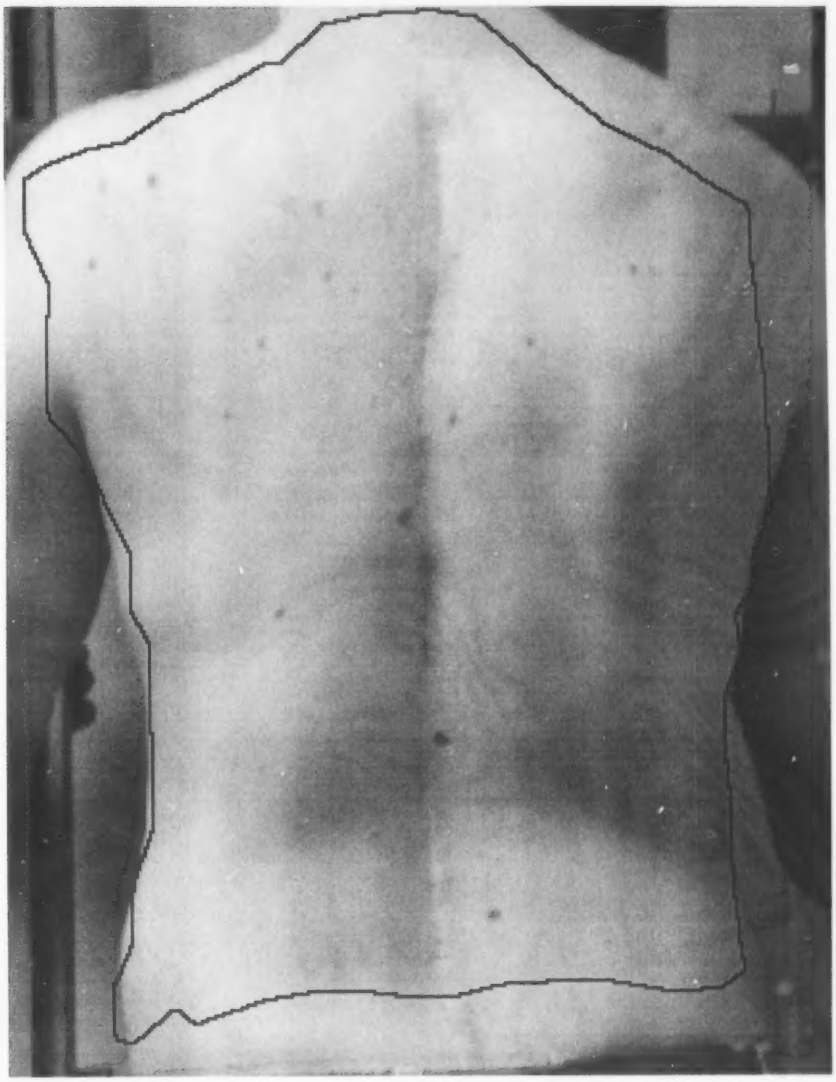

Figure 6.1: Boundary Accuracy: The computer generated boundary is shown in the black outline

the problems that occur when the point neighbourhoods cannot be determined, due to the lack of a uniform density of points. The next figure (II) illustrates the fully connected version of that same surface. This was done by increasing the allowed connection length so that the connections can span large holes (holes as seen in the first figure (I)). Note however, by increasing the allowed connection line length the outer border has begun to include regions which are not part of the back surface (thin triangles around the outer boundary of the mesh), and therefore, lowered the accuracy of the outer boundary. This clearly shows the importance of having a close to uniform density of points so the local neighbourhoods can be defined automatically.

The next two figures (III and IV) are data sets obtained using structured light. The mesh generation is both complete and remains within the boundaries of the data set. The third figure (III) was obtained from a periodic dot pattern projection and the fourth was every fourth point from a line rasterstereography data set. The periodic projection of light onto the back surface ensures a relatively uniform density and thus local neighbourhood information is derived which creates all valid connections.

The planar triangles used in the triangulation process do not try to predict or interpolate accurately the surface coordinates between the individual data points. Rather, the triangles provide a solid looking surface making no assumptions about how the actual surface may undulate between data points. Accuracy can only be determined absolutely if data were taken from a mathematically defined object so that every intermediate value were known and could be compared with the planar triangulated surface. However in the absence of a mathematical data set representa-

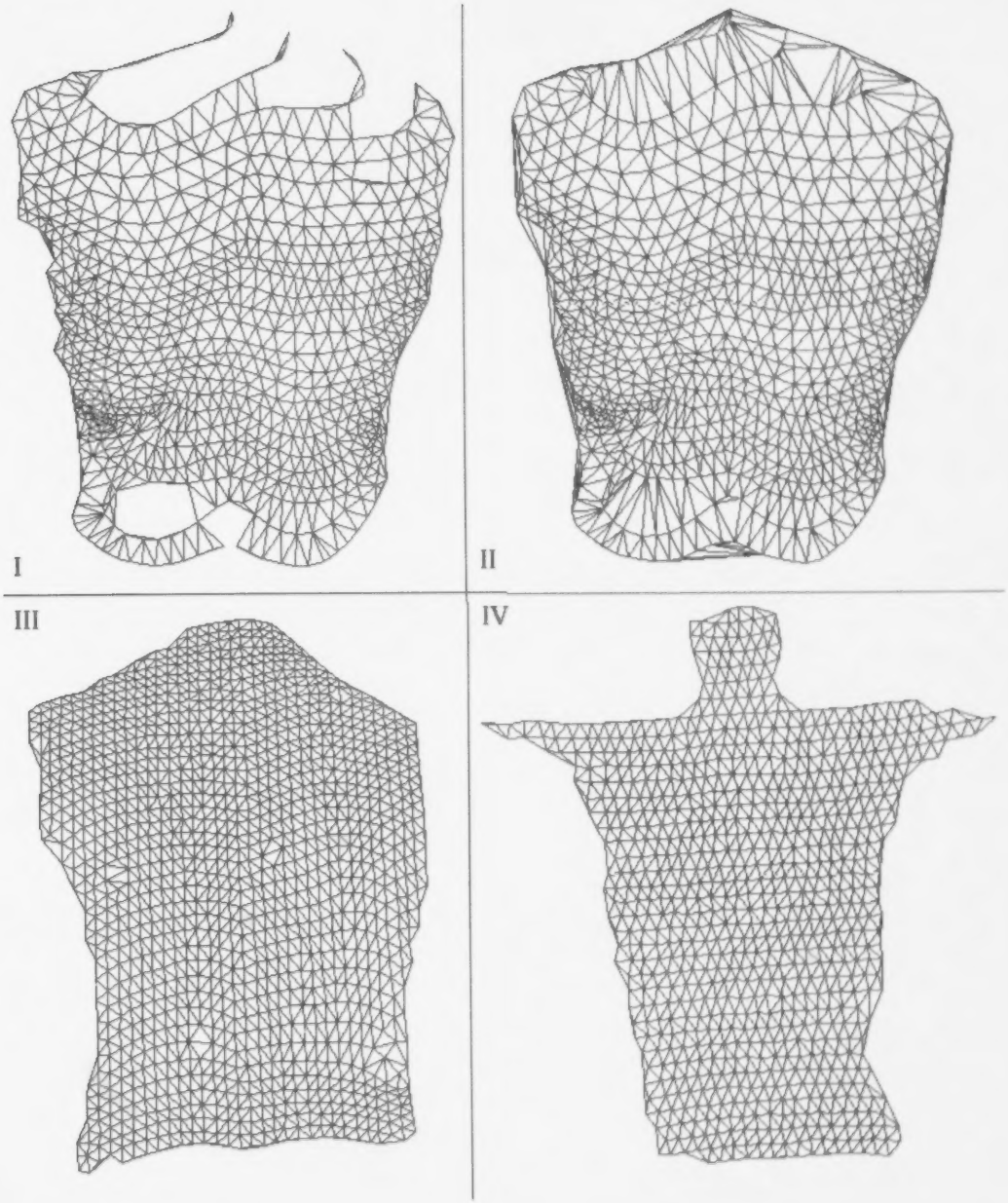

Figure 6.2: I) An inadequately sampled surface; II) The fully connected surface from I; III) Structured light data (dot pattern); IV) Structured light data (rasterstereography)

tive of a general back surface, a relative accuracy can be determined for the surface and it can be shown that the planar triangulation does not loose accuracy after the custom noise filtering process.

Half the points from two data sets were removed and both resulting sets of data were triangulated into respective surfaces. The result is two surfaces at approximately half the original data density. Using the assumption that the data left out of the surface was a good approximation to the exact surface, the distances of the "exact" data to the surface was determined for each of the two surfaces. The distance in the Z coordinate from each "exact" point to the surface was determined and the resulting deviation from the surfaces are given in table 6.1.

Surface Source	Exact Data Assumed	Standard Deviation	Root Mean Squared Error
Dot Projection Surface	1st half of original data	1.3mm	1.1mm
	2nd half of original data	1.7mm	1.3mm
Custom Filtered Dot Projection Surface	1st half of original data	1.2mm	0.95mm
	2nd half of original data	1.5mm	1.2mm
Rasterstereography Surface	1st half of original data	1.4mm	1.1mm
	2nd half of original data	1.4mm	1.1mm
Custom Filtered Rasterstereography Surface	1st half of original data	1.4mm	1.1mm
	2nd half of original data	1.3mm	1.0mm

Table 6.1: Surface Accuracy: Two surfaces with half the points missing and half assumed to be exact

The same was done for the surfaces after reducing the effects of noise. It was desired that the process of filtering would not decrease the accuracy of the resulting surfaces. The results in table 6.1 indicate that the relative accuracy is comparable to that of the triangulation of the original data set, when compared with the original

data points (assumed to be exact surface data points).

6.1.3 Accuracy - Effects of Noise

The last stage of the surface generation was to remove the effects of noise. The accuracy was measured as the uncertainty in the Z coordinate. As the surface was manipulated through filtering, the data points will be moved. The distance that each data point moves from its original position adds to the uncertainty found in the data acquisition stage. The three filters (local average, median and custom filter) were evaluated on how well they removed the effects of noise for a given accuracy. An desired, maximum uncertainty was set before the filtering was done and then the filters were evaluated qualitatively. The details of these results are provided in the next section on realism.

6.2 Realism

As with accuracy, realism was evaluated separately for each of the stages: boundary generation, triangulation, and reducing the effects of noise. For each stage the characteristics that were used to evaluate realism are given and the results are presented in a graphic comparison for the triangulation and noise portions.

6.2.1 Realism - Boundary

A boundary was considered realistic if it complies with the properties of the actual surface boundary. The actual boundary of the back surface creates a continuous

contour in two dimensions which outlines the back and follows the outlines of the neck arms and shoulders. How closely the computer generated boundary comes to the actual boundary was limited by the density and the area which the data covers. The area given by body structures outside the back have little or no coverage in the data sets. Therefore, arms and the neck region are cut off from being included in the boundary. The data points near the actual edge of the back are often missing due to limitations in the data acquisition process. At the edges of the back surface the relative angles between the projector and the reflecting surface areas are not adequate to allow the camera to detect the light pattern. Without the adequate detection of the light, no data points can be calculated. For this reason, all computer generated boundaries will have a smaller perimeter than the corresponding two dimensional boundary found on a photograph of the same surface. This is illustrated in the previously referenced figure 6.1.

A boundary's realism was also apparent in the way it holds to other boundary characteristics: non-overlapping, single polygon outline and continuous. The boundary was continuous to the degree in which the data was continuous. The first and second derivatives of the boundary are not continuous because the connecting lines (triangle edges) are linear. Whether or not a single polygon outline was created was determined by whether the surface was adequately sampled as seen in figure 6.2. Since the surface in figure 6.2 I) was not adequately sampled to describe the entire surface, disconnected bounding polygons create holes amidst the surface. This was also evident in figure 6.3 when clothing or other objects are included in the data

set, and they cannot be adequately sampled with structured light techniques. The property of non-overlapping was taken care of by the triangulation process in two dimensions. No triangles can overlap, therefore, no boundary line can overlap.

6.2.2 Realism - Triangulation

The triangulation process was meant to add a solid continuous surface appearance in three dimensions. The non-overlapping property of the surface was taken care of by triangulating in two dimensions and doing exhaustive testing for overlapping triangles. However, the continuity of the surface was not fully addressed by the simple triangle mesh, so other measures had to be taken to provide the illusion of a continuous surface. Rather than using an arbitrary triangulation, which produces triangles of arbitrary dimensions, a Delaunay triangle optimization stage was used to eliminate the effects of thin triangles (see figure 2.6). Two figures are now introduced which show the stages before and after triangle optimization for one data set. Figure 6.4 A) is before optimization and the lighting fluctuations are more pronounced (due to thin triangles) than in the optimized surface: figure 6.4 B).

The second way in which the illusion of continuity was achieved was through shading. Even though the first derivatives (slopes) of abutting triangles may instantaneously change at their meeting point the bilinear shading (Gouraud) creates the appearance of a smooth transition from one surface element to the next. Figures 6.5 A) and B) illustrate the flat shaded and Gouraud shaded surfaces.

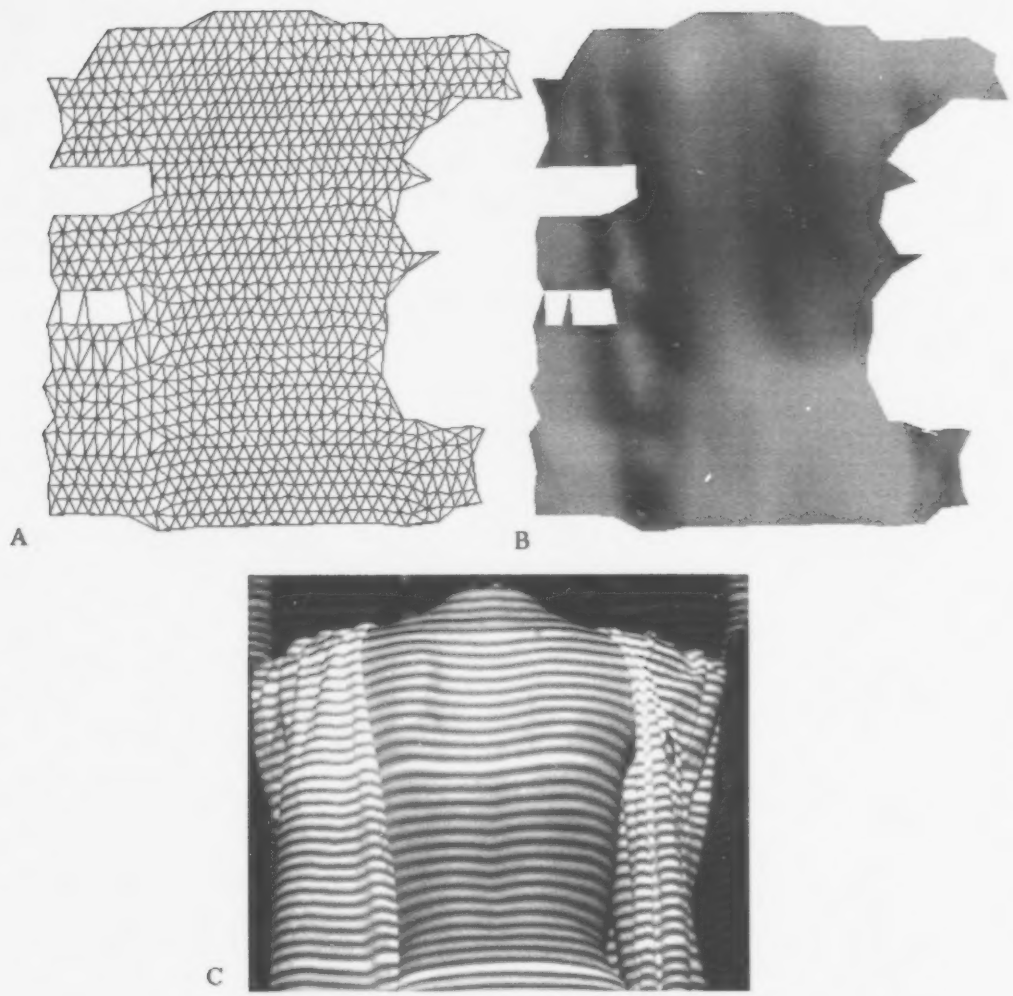

Figure 6.3: Triangulation of a back surface and surrounding clothing

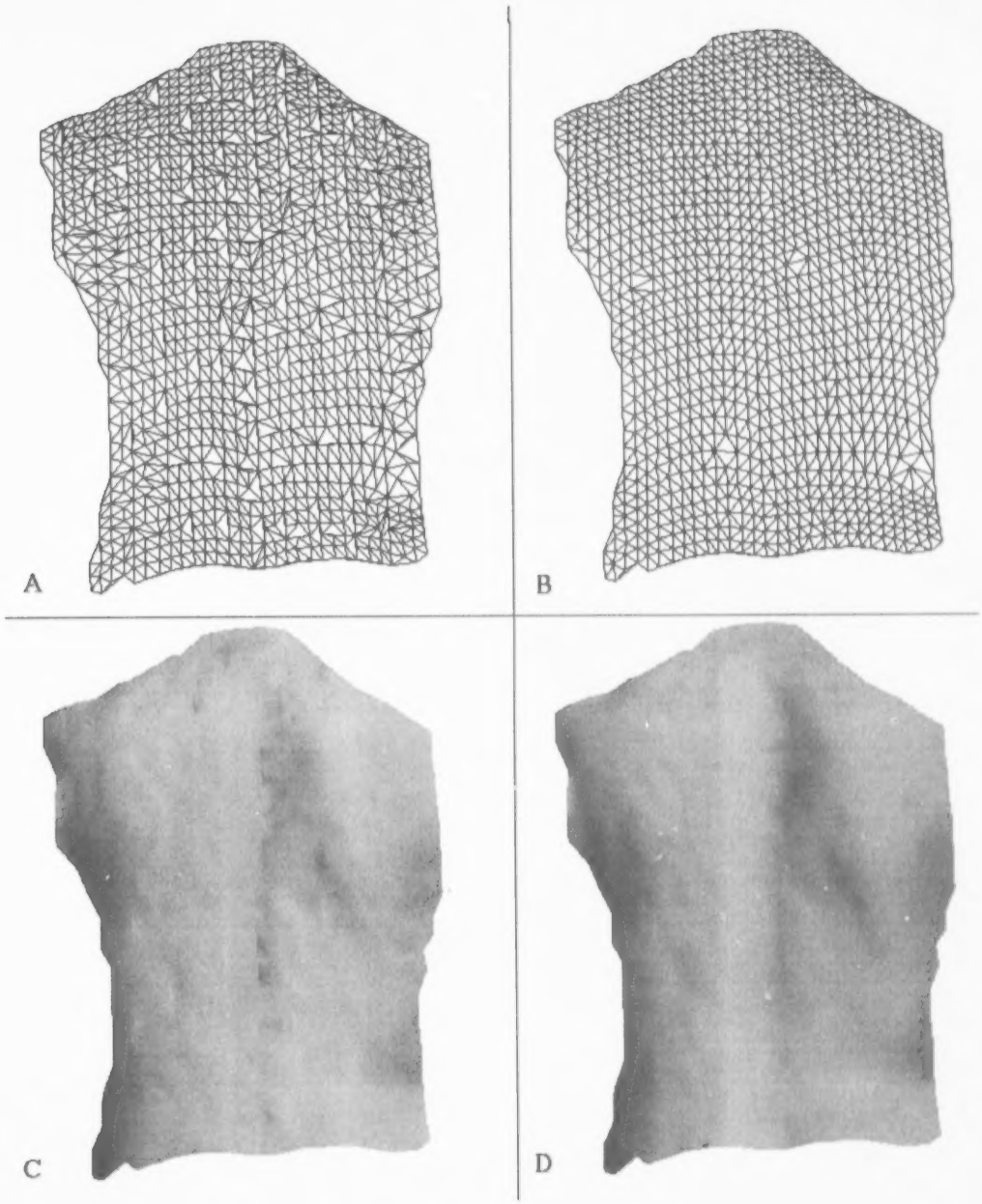

Figure 6.4: Triangle Optimization: A,C) Arbitrary triangle mesh (stage 1); B,D) Optimized triangulation (Delaunay triangle mesh)

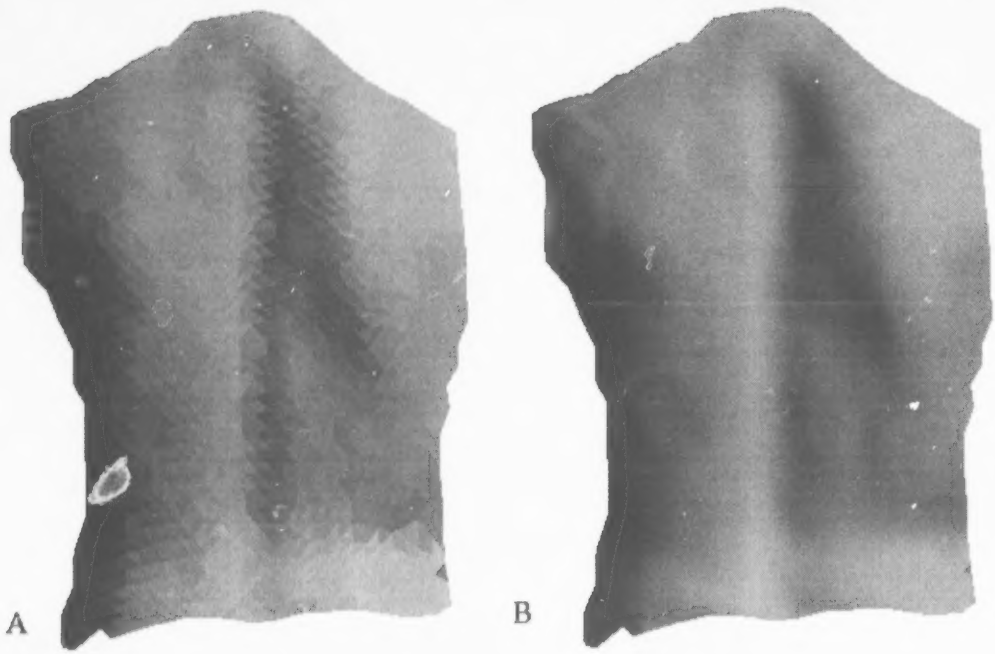

Figure 6.5: Triangle Shading:A) flat shaded triangles; B) Gouraud shaded triangles

6.2.3 Realism - Effects of Noise

The results for reducing the effects of noise is a simple visual comparison between the three filtered surfaces and the original surface. Each surface is Gouraud shaded and each of them was limited to moving any of the data points by more than $1.5mm$ in the Z coordinate. The local average and median filters required the surface to be resampled and the custom filter did not. The results are presented in figures 6.6 and 6.7.

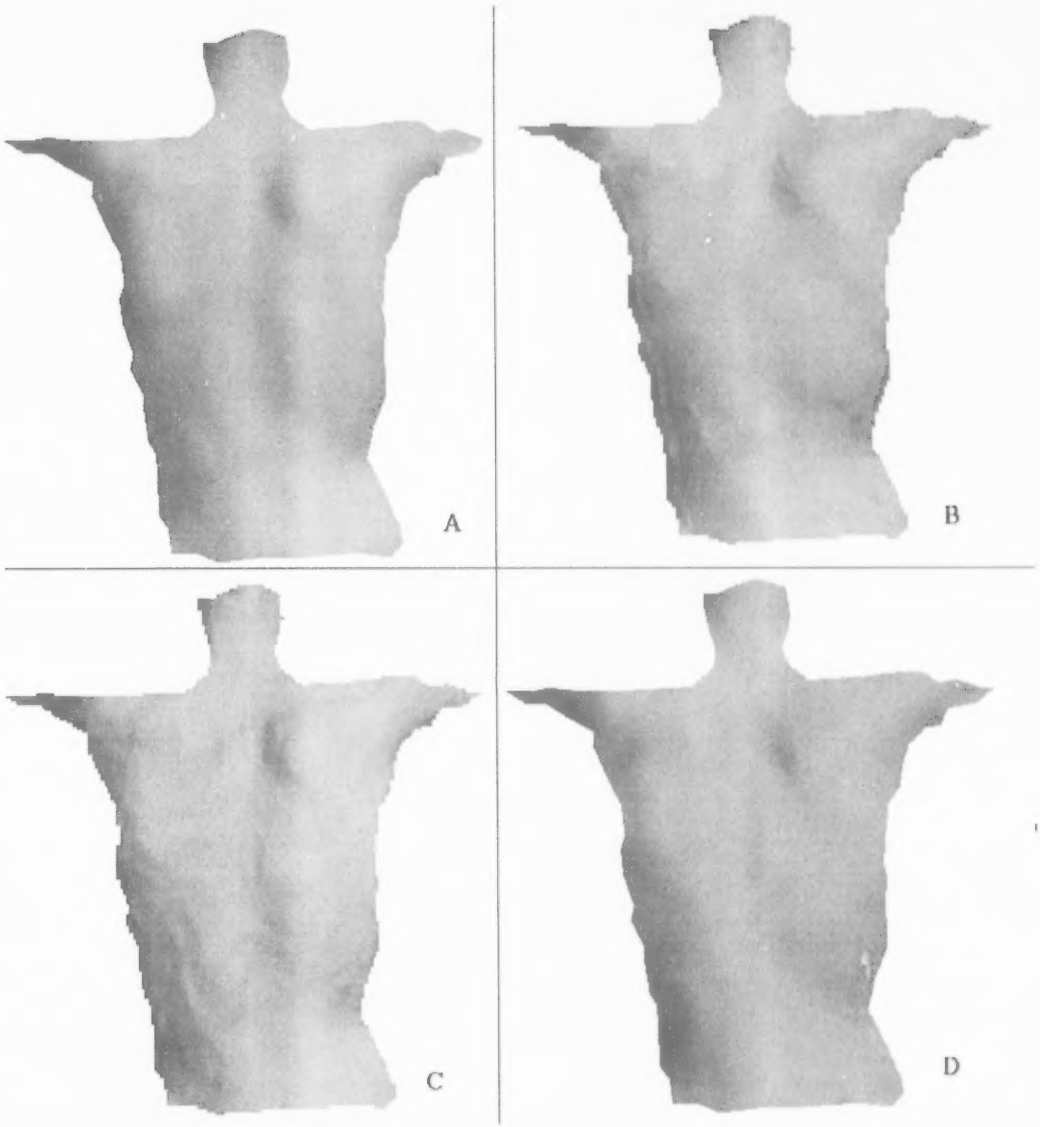

Figure 6.6: Back Surface I: A) Original surface; B) 9x9 local average filter, resampled surface; C) 9x9 median filter, resampled surface and D) Custom noise filter

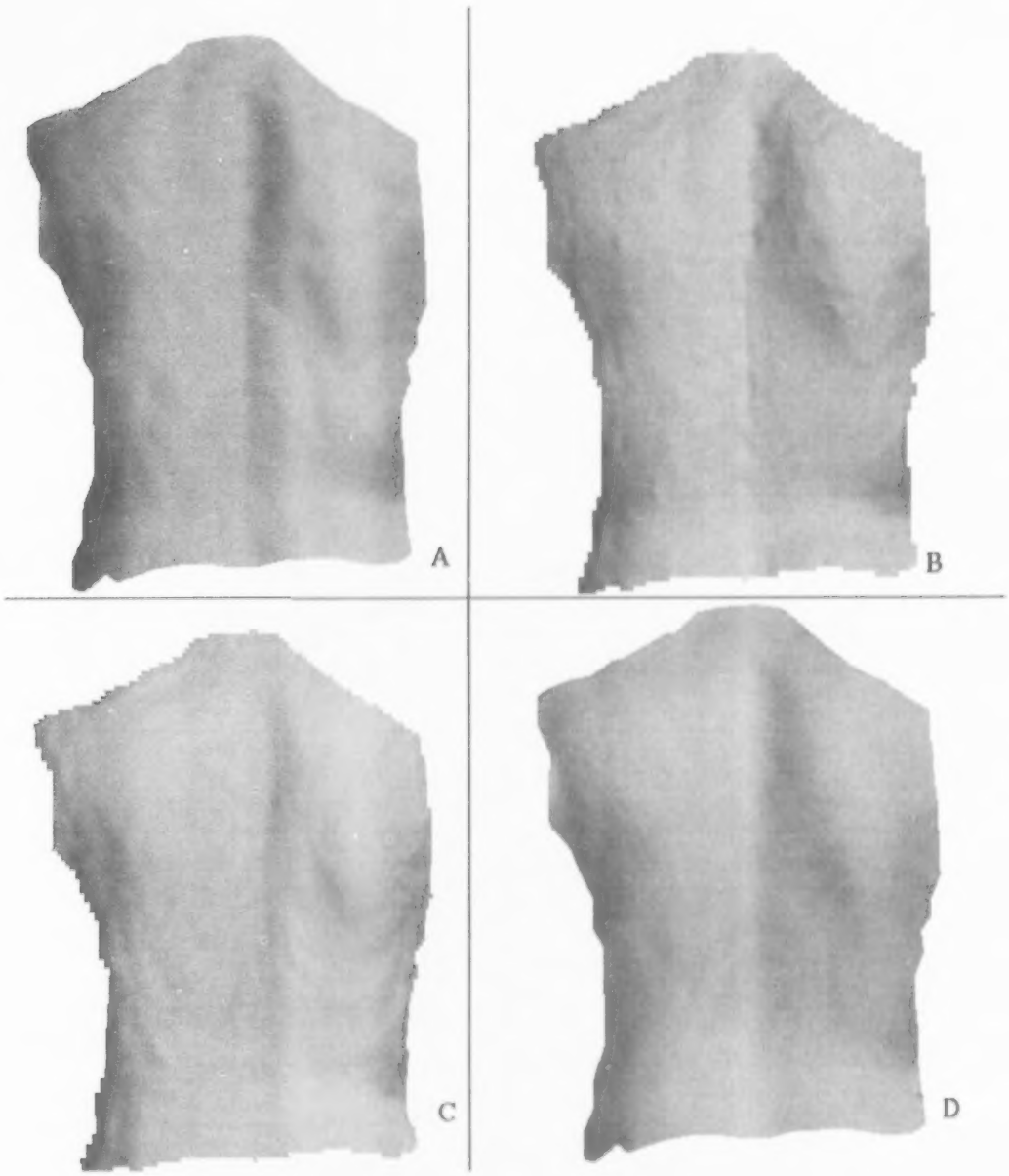

Figure 6.7: Back Surface II: A) Original surface; B) 9x9 local average filtered, resampled surface; C) 9x9 median filter, resampled surface and D) Custom noise filter

6.3 Speed

The amount of time it took to determine the approximate density of the data set and to apply the noise filters was on the order of one second for all of the data sets tested (less than 2000 original data points). The longest time spans occurred when generating the optimized triangle mesh and when rendering the triangulated mesh on the screen. For this reason, the results pertaining to speed of processing will be restricted to the triangulation process. All computer calculations were done by an RS6000 and the graphics were rendered by software and a graphics accelerator board. The details of the computer used are presented in chapter 1 under Computer Environment.

6.3.1 Speed - Triangulation

The choice of a triangle for the surfacing element minimizes the rendering time for the surface. All surfaces refresh within one second with up to ≈ 15000 triangles (≈ 7500 data points). All original back surface data will contain fewer triangles. However, this number is exceeded if a sufficiently small resampling period is used or the data set area is increased.

Although the triangulation time is not crucial to the performance of the display, it is presented here as time taken to preprocess the data for display. The three stages of triangulation have the orders as given in table 6.2.

The experimental timing graphs for the total triangulation process for data sets with less than 5000 points is given in figure 6.8. The experimental connection vs

Stages	Theoretical Order	Experimental Order
Initialization of data structures	O(n)	$O(n^2)$
Triangulation into a connected mesh	O(n)	O(n)
Triangle Optimization	O(n)	n/a

Table 6.2: Triangulation Algorithm Processing Time (Orders): Triangle optimization was not assessed experimentally, partially due to the extremely short processing times

initialization graph is shown in figure 6.9. The graph of total triangulation time is very close to linear, with a slight tendency to $O(n^2)$. The reason for the upturn in figure 6.8 can be seen in figure 6.9. The initialization time begins to increase in $O(n^2)$ and becomes a significant amount of the processing time after $\approx$ 5000 points. A limited number of data points were taken outside the reasonable range of 0-5000 points. These indicate that the initialization time will be equal to the triangulation time somewhere between 30 000 and 40 000 points. The possible reasons for this will be explored in the conclusions.

6.4 Automation

The entire process from beginning to end (boundary detection to display) requires the input of two items. First the Cartesian coordinates of each point in the data set must be known. This data set has no restrictions placed upon its ordering. The second input is the error in the data acquisition system, or the amount of uncertainty that is allowed to be introduced by the noise filtering process. With these two items, any data which is sufficiently uniform in density can be rendered in three dimensions. Up until now the images that have been presented have been of a single viewpoint

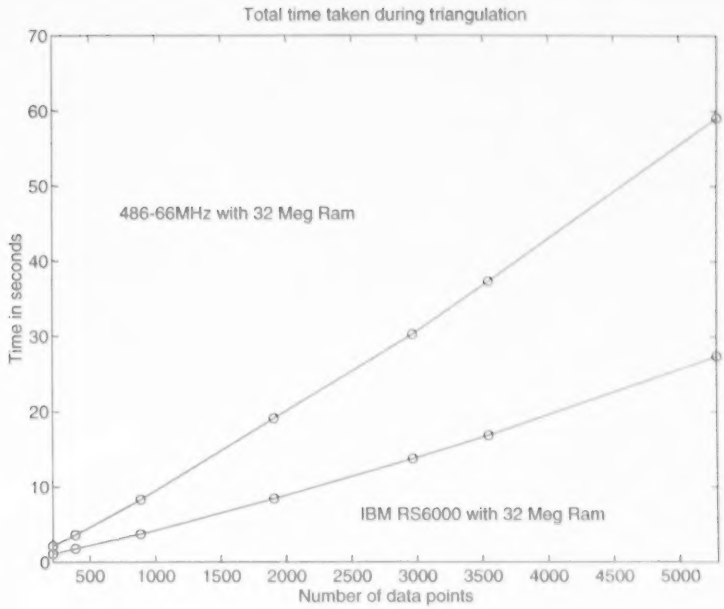

Figure 6.8: Total Triangulation Time vs Number of Points: on two different computer processors

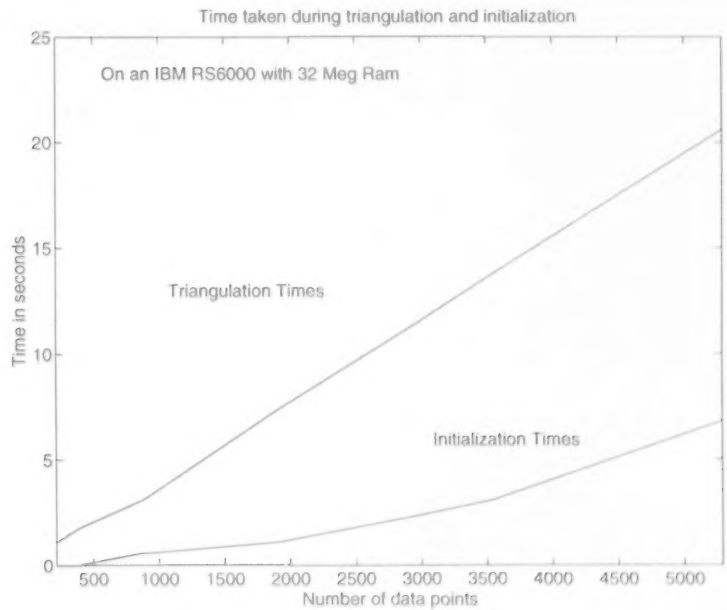

Figure 6.9: Triangulation and Initialization Graph: on an RS6000 with 32 Meg Ram

of the surface. This was to keep all images consistent. Figure 6.10 is shown here to illustrate the three dimensional nature of the back surfaces .

Figure 6.10: Back Surface I: different view points

Chapter 7

Conclusions and Future Work

The objective was to automate a process of transformation which will convert discrete three dimensional data into a realistic and accurate representation of the human back. In doing so, the process was to minimize the amount of user interaction and errors due to data transformation, and maximize the speed of the computer representation and manipulation.

It was found that in order to portray a realistic back surface on the computer screen the surfacing element would need to be easily rendered. This would be necessary if the rendering speed was to be under a couple of seconds for data sets of over 1000 points (which is typical for back surface data). The triangle is the smallest surfacing polygon and renders the fastest of all solid surfacing elements which could create a surface. A triangle mesh was therefore used to generate a solid looking surface.

To ensure full connectivity and valid connections when creating the triangle mesh it was necessary to derive some approximation to the density of the data set (when back projected onto the X-Y plane). An appropriate approximation can be derived

by taking sample areas within the boundaries of the data and choosing the most common density of points. A line length which will make all valid triangle connections is derived from this density approximation. When calculating an appropriate line length the non-uniformities in the density must be taken into consideration so that all valid connections can be made across the entire back surface leaving no holes in the data set.

It was found that by creating all valid triangle connections, an appropriate boundary was automatically created for the data set. This eliminated the need to explicitly know the type of object and its inherent shape. The connectivity of the surface relies on adequate sampling of the surface. Without adequate sampling it has been shown (see figure 6.2 I) that the connectivity that is intuitive for the shape of a back surface is not achieved. It is also shown in part II of figure 6.2 that the relaxation of the connectivity properties causes connections to be made around the outside of the data set, encompassing areas that are not indicative of the underlying actual surface: thus reducing both realism and accuracy of the representation.

Once the solid connected surface of triangles was created it was necessary to interpolate the shading across the surface of each triangle to improve the illusion of a continuous surface. The bilinear interpolation provided by Gouraud shading was the best solution for shading, because the calculations could be performed in the graphics acceleration hardware. The noise present in the data was the last major hurdle to creating a realistic surface.

The three filters that were tested for effectiveness on removing the effects of noise

were tested for constant error tolerances. The custom filter, which acts as a local average filter on the triangle mesh, had significantly better results than either the standard local average or median filters. The reason for this is that in order to implement the local average and median filters it is necessary to know the ordering of the points in terms of a periodic grid of data points. In the process of resampling error is introduced which is not introduced during the custom filtering. Since the error tolerances are set values, some of that tolerance was used by resampling, and left less tolerance for the filtering process.

The custom filtering resulted in a triangulated back surface which had a relative accuracy less than or equivalent to that of the planar triangulation of the original data set before filtering. The results shown in table 6.1 show deviations from the "exact" surface data which tend towards values within the actual data acquisition accuracy of approximately $\pm$2mm.

The entire process from the raw original data to a three dimensional computer representation of the surface is shown for one data set in figure 7.1. The amount of processing time for a data set of less than 2000 points is under 20 seconds. The on screen rendering of the same surface is less than one second between refreshes when turning the object in three dimensions.

7.0.1 Future Work

As the number of data points increases above 20 000 the initialization time becomes the dominant factor in the overall performance of the triangulation. The order of the

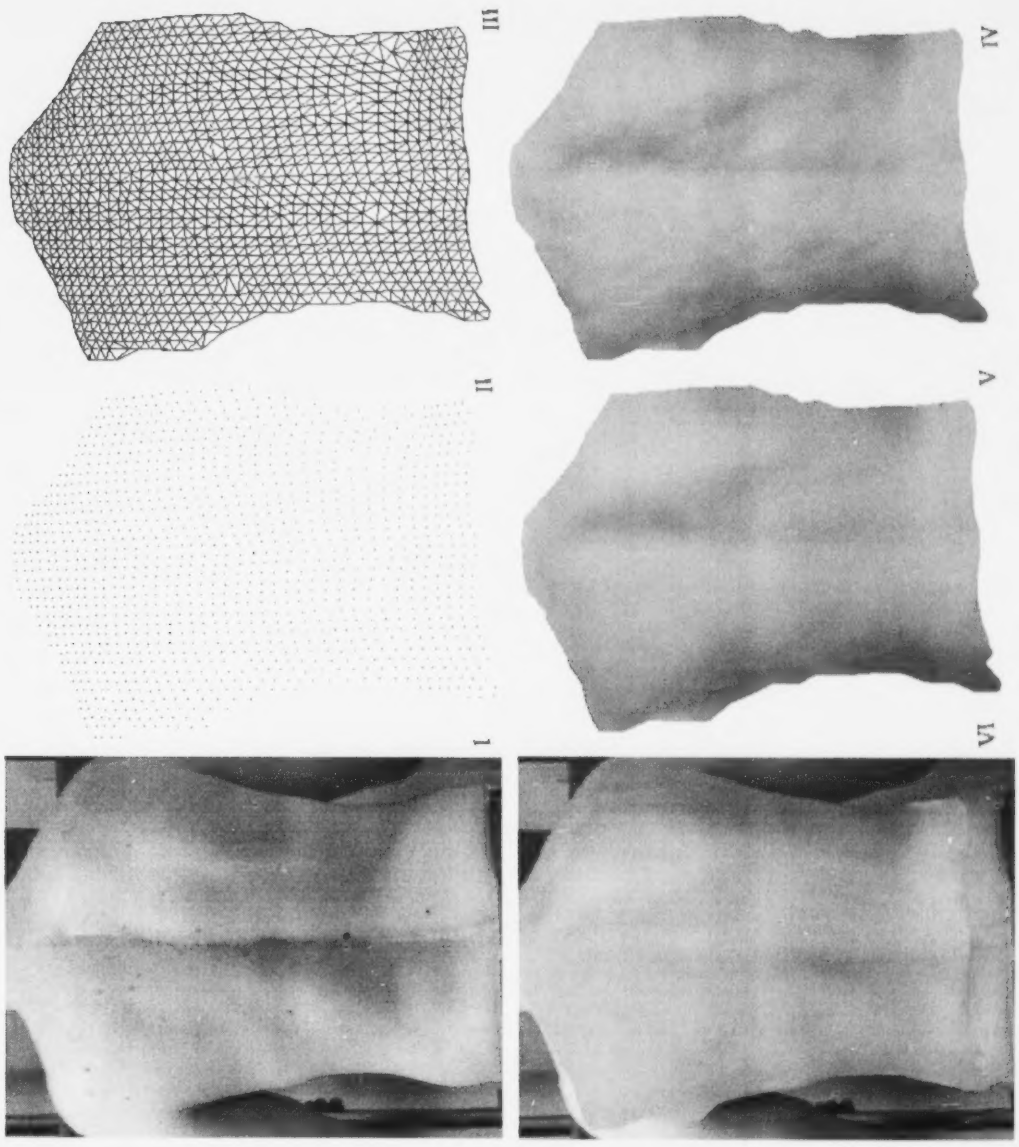

Figure 7.1: Graphical Summary: I) Photograph of the patient's back; II) Original data set in the X-Y plane; III) triangle mesh; IV) Shaded surface; V) Removal of the effects of noise; VI) The computer surface superimposed on the patient's photograph

triangulation initialization should be reducible to $O(n)$. The reason for the higher order was the allocation of memory becomes inefficient as the number of points increases. Intuitively, the number of points should not affect the time it takes to allocate memory, but the same memory allocations take longer the more points that are added to the initialization structure. Since precisely the same amount of work is theoretically required to add each data point to the structure, it is desirable to reduce the actual amount of time it takes to linear. This would then make the entire triangulation program linear and it more easily applied to large data sets for this or other applications. Experiments in allocating larger areas of memory at a time resulted in gains in speed of three times. This is still not linear and therefore more work on this problem is required.

If the data set is too large for screen representation then techniques for triangle decimation [27][23] and multiple scaling of surface data [25] can be used to both reduce the number of surface elements used to render a surface and consequently speed up the rendering time.

Other structured light data sets or laser range finders will have very similar data sets and the data transformations presented will work with this data. The level of noise in laser range finding may not require the noise reduction stage, but the triangulation and boundary detection information would be of some use. The type of surfaces being imaged and the projector (laser, or projected structured light) and camera angles may require data rectification in order to calculate the appropriate densities for the data set. This was described in the boundary implementation section.

A resampled (regular grid) version of the data can be used to quantify and monitor changes in the surface shape [21]. By using surface curvatures along with other rotation and scale independent measures the changes in shape can be successfully quantified. This is the next step in the development of the analysis program being developed for the Glenrose Rehabilitation Hospital.

Bibliography

[1] S. Aubry and Vincent Hayward. *Recursive Decomposition of Free-Space From Boundary Points.* McGill University, 1987.

[2] Francis J. M. Schmitt Brian A. Barsky and Wen-Hue Du. An adaptive subdivision method for surface-fitting from sampled data. *SIGGRAPH*, 20(4):179–187, August 1986.

[3] J-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Trans. on Graphics*, 3(4):226–286, October 1984.

[4] Tom Duff. Smoothly shaded renderings of polyhedral objects on raster displays. *SIGGRAPH*, 4(4):270–275, 1979.

[5] Dugeon and Mercereau. *Multidimensional Signal Processing.* Prentice Hall, Englewood Cliffs, New Jersey, 1984.

[6] D. Hill V.J. Raso N.G. Durdle and A.E. Peterson. Designing a video based technique for trunk measurement. *International symposium on 3-D Scoliotic Deformity*, pages 157–161, 1992.

[7] Todd Elvins. A survey of algorithms for volume visualization. *Computer Graphics*, 26(3):194–201, 1992.

[8] A. Ekoule et al. Reconstruction and display of biomedical structures by triangulation of 3d surfaces. *SPIE: Science and Engineering of Medical Imaging*, 1137:106–113, 1989.

[9] Henry Rusinek et al. Three-dimensional rendering of medical images: surface and volume approach. *Image Capture and Display*, 1091:204–211, 1989.

[10] Foley. *Principles in Computer Graphics: Second Edition*. Addison-Wesley Publishing Company, 1990.

[11] Robert J. Fowler and James J. Little. Automatic extraction of irregular network digital terrain models. *In ACM*, pages 199–205, 1979.

[12] W. Frobin and E. Heirholzer. Transformation of irregularly sampled surface data points into a regular grid and aspects of surface interpolation, smoothing and accuracy. *SPIE: Biosterometrics*, 602:109–115, 1985.

[13] C.M. Gold. *Triangulation-Based Terrain Modelling-Where are we now?* Autocart, 1979.

[14] P.J. Green and R. Sibson. Computing dirichlet tessellations in the plane. *The Computer Journal*, 21:168–173, 1977.

[15] Mark Hall and Joe Warren. Adaptive polygonalization of implicitly defined surfaces. *IEEE Computer Graphics and Applications*, pages 33–42, November 1990.

[16] Whoi-Yul Yura Kim. *Efficient strategies for reducing the size of saerch space in 3-d object recognition and pose estimation.* PhD thesis, University of Texas, 1989.

[17] B.A. Lewis and J.S. Robinson. Triangulation of planar regions with applications. *The Computer Journal*, 21(4):324–332, 1978.

[18] Richard Liu. Three dimensional reconstruction of trunk surface using structured light. Master's thesis, University of Alberta, 1995.

[19] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3-d surface reconstruction algorithm. *Computer Graphics*, 21(4):161–169, 1987.

[20] Hugues Hoppe Tony DeRose Tom Duchamp John McDonald and Werner Stuetzle. Surface reconstruction from unorganized point. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26(2):71–77, July 1992.

[21] Michael F. Polak. Quantifying scoliotic shape using surface curvature. Master's thesis, University of Alberta, December 1995.

[22] R. Sibson. Locally equiangular triangulations. *The Computer Journal*, 21(3):243–245, 1978.

[23] Marc Soucy and Denis Laurendeau. Multi-resolution surface modeling from multiple range views. *IEEE: Computer Vision and Pattern Recognition*, pages 348–353, 1992.

[24] Ulf Tiede. Investigation of medical 3d-rendering algorithms. *IEEE Computer Graphics & Applications*, pages 41–53, March 1990.

[25] Greg Turk. Re-tiling polygonal surfaces. *Computer Graphics*, pages 55–64, July 1992.

[26] R.C. Veltkamp. *Closed Object Boundaries from Scattered Points*. Springer Verlag, Berlin, 1994.

[27] William J. Schroeder Jonathan A. Zarge and William E. Lorensen. Decimation of triangle meshes. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26(2):65–70, July 1992.

[28] Zixi Zhang. Surface modelling of trunk deformity. Master's thesis, University of Alberta, 1993.